Feedback control for sparsity based regularization parameter selection in X-ray tomography

University of Helsinki

Juho Rimpeläinen

November 28, 2017

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of Science	Department of mathematics and statistics

Tekijä — Författare — Author
Juho Rimpeläinen

Työn nimi — Arbetets titel — Title

Feedback control for sparsity based regularization parameter selection in X-ray tomography

Oppiaine — Läroämne — Subject
Applied mathematics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	December 2017	55

Tiivistelmä — Referat — Abstract

X-ray computed tomography is an imaging method where the inner structure of an object is reconstructed from X-ray images taken from multiple directions around the object. When measurements from only a few measurement directions are available, the problem becomes severely ill-posed and requires regularization. This involves choosing a regularizer with desirable properties, as well as a value for the regularization parameter.

In this thesis, sparsity promoting regularization with respect to the Haar wavelet basis is considered. The resulting minimization problem is solved using the iterative soft thresholding algorithm (ISTA). For the selection of the regularization parameter, it is assumed that an *a priori* known level of sparsity is available. The regularization parameter is then varied on each iteration of the algorithm so that the resulting reconstruction has the desired level of sparsity. This is achieved using variants of proportional-integral-derivative (PID) controllers.

PID controllers require tuning to guarantee that the desired sparsity level is achieved. We study how different tunings affect the reconstruction process, and experiment with two adaptive variants of PID controllers: an adaptive integral controller, and a neural network based PID controller. The two adaptive methods are compared to each other, and additionally the adaptive integral controlled ISTA is compared to two classical reconstruction methods: filtered back projection and Tikhonov regularization. Computations are performed using both real and simulated X-ray data, with varying amounts of available measurement directions.

The integral control is shown to be crucial for the regularization parameter selection while the proportional and derivative terms can be of use if additional control is required. Of the two adaptive variants, the adaptive integral control performs better with respect to all measured figures of merit. The adaptive integral controlled ISTA also outperforms the classical reconstruction methods both in terms of relative error and visual inspection when only a few measurement directions are available. The results indicate that variants of the PID controllers are effective for sparsity based regularization parameter selection. Adaptive variants are very end user friendly, avoiding the manual tuning of parameters. This makes it easier to use sparsity promoting regularization in real life applications. The PID control allows the regularization parameter to be selected during the iteration, thus making the overall reconstruction process relatively fast.

Avainsanat — Nyckelord — Keywords
Tomography, sparsity, wavelets, regularization, feedback control, PID controller

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Matemaattis-luonnontieteellinen	Matematiikan ja tilastotieteen laitos

Tekijä — Författare — Author
Juho Rimpeläinen

Työn nimi — Arbetets titel — Title

Feedback control for sparsity based regularization parameter selection in X-ray tomography

Oppiaine — Läroämne — Subject
Soveltava matematiikka

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
pro gradu	Joulukuu 2017	55

Tiivistelmä — Referat — Abstract

Röntgentomografia on kuvantamismenetelmä, jolla pyritään selvittämään objektin sisärakenne eri suunnista otettujen röntgenkuvien perusteella. Kun käytettävissä on kuvia vain hyvin pienestä määrästä suuntia, tomografiaongelma on äärimmäisen huonosti asetettu ja sen ratkaisu vaatii regularisointia. Regularisointi koostuu sopivan regularisoijan ja regularisointiparametrin arvon valinnasta. Tämä tutkielma käsittelee tapausta, jossa halutaan regularisoida rekonstruktion harvuutta Haarin aallokekannan suhteen. Se johtaa minimisaatio-ongelmaan, joka ratkaistaan iteratiivisella pehmeää kynnystystä käyttävällä algoritmilla (iterative soft thresholding algorithm eli ISTA). Regularisointiparametri valitaan olettamalla että käytettävissä on *a priori* tunnettu harvuustaso, eli luku joka kertoo kuinka suuri osuus objektia kuvaavista aallokekertoimista on erisuuria kuin nolla, ja säätämällä regularisointiparametria iteraation aikana niin, että rekonstruktion harvuus saavuttaa kyseisen tason. Tätä varten käytämme variaatioita proportional-integral-derivative-säätimestä (PID-säädin).

Jotta haluttu harvuustaso saavutetaan tulee säädin virittää sopivasti. Tutkimme eri virityksien vaikutusta rekonstruktioprosessiin ja erityisesti käsittelemme kahta adaptiivista säädinvarianttia parametrin valinnassa. Vertailemme näitä kahta varianttia, adaptiivista integraalisäädintä ja neuroverkkoihin perustuva PID-säädintä, toisiinsa. Lopuksi vielä vertaamme adaptiivisella integraalisäätimellä säädettyä ISTA:a kahteen klassiseen rekonstruktioalgoritmiin: suodatettuun takaisinprojektioon (filtered back projection, FBP) ja Tikhonov regularisointiin. Kokeissa käytetään sekä aitoa että simuloitua röntgendataa sekä verrattain tiheällä että harvemmalla mittauskulmien jakaumalla. Integraalisäätö on osoitettiin tärkeäksi regularisointiparametrin valinnassa, kun taas kahta muuta termiä voidaan hyödyntää tarpeen vaatiessa. Adaptiivisista säätimistä adaptiivinen integraalisäädin osoittautui kaikin kriteerein paremmaksi. Adaptiivisella integraalisäätimellä säädetty ISTA myös päihitti molemmat klassiset menetelmät sekä suhteellisen virheen että visuaalisen arvioinnin suhteen harvan datan tapauksessa.

Tulokset osoittavat että eri PID-säädinvariantit voivat toimia regularisointiparametrin valinnassa. Adaptiiviset säätimet ovat hyvin käyttäjäystävällisiä, koska ne eivät vaadi manuaalista parametrien säätöä. Lisäksi säätimet ovat verrattain yksinkertaisia, joten niiden soveltaminen eri tilanteissa on helppoa. PID-säätimet mahdollistavat regularisointiparametrin valinnan algoritmin suorituksen aikana, tehden näin koko rekonstruktioprosessista verrattain nopean.

Avainsanat — Nyckelord — Keywords
Tomografia, harvuus, aallokkeet, regularisointi, takaisinkytkentä, PID-säädin

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

Chapter 1

Introduction

X-ray computed tomography (CT) is a problem of finding the inner structure of an object from a set of X-ray images taken from multiple measurement directions around the object. The significance of medical CT-imaging is clear: approximately 70 million CT-scans are performed annually in the US alone. In this thesis we consider the sparse angle tomography, which allows faster scanning and reduced radiation dosage, using sparsity promoting regularization in the Haar wavelet basis. The arising minimization problem is solved using the iterative soft thresholding algorithm and the regularization parameter is selected using a simple feedback control system.

While the basics of CT imaging, the Radon transform and its inverse, have been known for decades, there are still problems to be solved. The classical reconstruction methods only work when the measurements from a large number of directions are known, however in medical imaging applications it is often desirable to minimize the radiation dosage of the patient being scanned. Therefore measurements from only a few directions can be obtained. Incomplete data results in an *ill-posed inverse problem* which means that no unique or stable solution to the problem exists. This means that solving the problem requires some form of regularization. For more information on inverse problems related to CT see [1]. In recent history sparsity promoting regularization has gained considerable attention, see for example [2] [3][4] [5]. The idea of sparsity promoting regularization is that the reconstructed object is assumed to be sparse: it can be expressed in terms of finite number of nonzero coefficients in some suitable basis. If w_j are the coefficients describing the function f, the regularization problem can then be formulated as minimization of the functional

$$F = \|Af - m\|_2^2 + \lambda \sum_{j=1}^{N} |w_j|, \tag{1.1}$$

where A is a linear map describing the measurement process, m is the measurement and λ is the regularization parameter. The regularization parameter λ then controls the level of sparsity of

the reconstructions; that is the number of nonzero coefficients w_j. The choice of regularization parameter is a difficult problem for which many approaches exist.

In this thesis we use the iterative soft thresholding algorithm introduced in [2] with the Haar wavelet basis to obtain sparse 2D tomographic reconstructions from a small number of uniformly distributed measurement directions. The regularization parameter, or in our case the thresholding parameter, is chosen by assuming *a priori* known ratio of nonzero wavelet coefficients of the target function and applying proportional-integral-derivative (PID) controller to change the parameter during iteration and force the ratio of nonzero coefficients of the reconstruction to the desired level. This approach is related to the method introduced in [6] and its variant [7], however these methods require computations using multiple values of the regularization parameter and are therefore time consuming. Our method allows the parameter to be selected online with simple rules. This makes sparsity based regularization parameter selection possible for large scale problems. The idea of varying the regularization parameter in iterative soft thresholding is not new. This approach is closely related to continuation schemes, where a sequence of problems with decreasing regularization parameter are solved and the solution to previous problem is used as a warm start for the next problem. For a few examples of these sorts of methods see [8] [3] [4]. This sort of method could be useful in medical imaging where estimates for the desired sparsity levels could be computed from a set of previously obtained full data CT reconstructions.

Feedback controllers work by taking the output of the process being controlled y, comparing it to the desired level y_d and changing the controlled parameter, such as the position of a dial, accordingly. PID type controllers are some of the most widely used controllers for industrial processes [9]. The value of the controlled parameter of a system is obtained as the output of the PID controller which is of the form

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d e'(t), \tag{1.2}$$

where $K_p, K_i, K_d \in \mathbb{R}$ are the tuning parameters and $e = y(t) - y_d$. The process of choosing suitable parameters K_p, K_i and K_d is called tuning. Different approaches for tuning have been proposed over the decades: most well known being the Ziegler-Nichols method [10]. We use two simple adaptive variants of PID controllers which automatically tune themselves: the adaptive integral controller [11] and the neural network based PID controller [12].

The main goal of the thesis is to demonstrate that simple feedback control such as the variants of PID controllers could be used for thresholding parameter selection. To achieve this we will test different PID settings to demonstrate how each parameter affects the evolution of the ratio of nonzero coefficients of the iterates. We will test the two adaptive methods with two different datasets: the simulated and real X-ray projection data. The adaptive methods are then compared by taking a look at the number of iterations reconstructions took, the error of the simulated case and the effect on the evolution of the ratio of nonzero wavelet coefficients.

The thesis is structured as follows. In chapter 2 we discuss the X-ray computed tomography. We derive the continuous and discrete measurement models and discuss the inversion of the continuous case when complete projection data is known. We also briefly discuss the ill-posedness of CT imaging. In chapter 3 we discuss wavelets, first the basics in the continuous case, and then the discrete case considering specifically the multiresolution approach. We also give the wavelet transforms needed later in following chapters. In the chapter 4 the iterative thresholding algorithm and the parameter selection using PID controllers are discussed. We give some basic infomation on PID controllers and derive the equations describing the discrete time controllers. Additionally tuning of PID controllers is covered briefly and the two adaptive methods are presented. We discuss the convergence of our method only briefly, as it is out of the scope of this thesis. In chapter 5 we introduce the datasets used when testing the algorithms and some simple measurements used to asses the effectiveness of the different PID variants. Chapter 6 contains our numerical experiments which are discussed in Chapter 7.

Chapter 2

Computed tomography as a linear inverse problem

As the X-ray beam travels through material some of its energy is lost as the photons encounter electrons in their path. Some materials block more radiation than others, and thus an X-ray projection image is formed. X-ray tomographic imaging uses these projections to reconstruct the inner structure of an object modeled as an attenuation function. In this thesis we will consider the two dimensional case of tomography, in this case the measurements from individual measurement directions are then one dimensional. In this chapter we derive the measurement model for tomographic imaging, and discuss the ill-posedness of the inverse problems involved in CT. This chapter is based mainly on [1] and [13] as well as [14].

2.1 The measurement model

Consider the case of X-rays, or any other penetrating rays, passing through an object. The intensity of the beam is known before entering the object and on the other side of the object the intensity is measured with a detector. As the ray passes through the object, some photons encounter electrons of the material and are absorbed, thus lowering the measured intensity. This is called attenuation. Different chemical elements have different attenuation properties making some materials more or less visible in the X-ray images than others.

Let us model the inner structure of the object by how much different areas attenuate the X-rays. The function $f : [a, b] \times [a, b] \to \mathbb{R}$ is a non-negative attenuation coefficient function, that is the value $f(x_1, x_2) \geq 0$ tells us how much the intensity of the ray is lowered as it passes through the point (x_1, x_2). A ray passing through a small distance Δx at the point $x = (x_1, x_2)$ loses a portion of its intensity. If I denotes the intensity of the beam, and ΔI denotes the

change in the beam intensity we have a relative intensity loss of

$$\frac{\Delta I}{I} = -f(x)\Delta x. \tag{2.1}$$

Now if we consider the beam traveling through the object along some line L from above we get a differential equation

$$\frac{dI}{dx} = -f(x)I. \tag{2.2}$$

Solving for I and using the initial condition $I(0) = I_0$ one has

$$\int_L f(x)ds = \log\left(\frac{I_0}{I_d}\right) = \log(I_0) - \log(I_d), \tag{2.3}$$

where I_0 is the beam intensity from the calibration measurement, and I_d is the detected intensity. It is worth noting that above model does not take the energy dependence of attenuation into account: different frequencies of radiation pass through materials easier than others. Since typical X-ray sources produce multispectral beams this can lead into a problem called *beam hardening*.

The idea behind CT imaging is to combine measurements from multiple angles to recover the inner structure of the object. In our case this means recovering a two dimensional slice of the object at some height. There are multiple ways to choose the lines over which the measurements are taken. This results in different *beam geometries*. The simplest beam geometry is called *parallel beam geometry*; all measurements from a single angle are taken with parallel beams. Another, more practical beam geometry is *fan beam geometry*. Here a point-like X-ray source emits a fan like set of beams which are measured with a detector plate.

Since tomographic data requires measurements from multiple angles, we need the definition of *Radon transform*, which maps the $\mathbb{R}^2$ functions to the sets of their line integrals, or more generally $\mathbb{R}^n$ functions to their integrals over hyperplanes.

Definition 2.4 (Radon transform)**.** Let $\boldsymbol{\theta} \in S^{n-1}$, and $s \in \mathbb{R}$. The Radon transform of function f is defined as

$$\mathcal{R}f(s, \boldsymbol{\theta}) = \int_{\boldsymbol{x}\cdot\boldsymbol{\theta}=s} f(\boldsymbol{x})d\boldsymbol{x}_L, \tag{2.5}$$

if above integrals exist.

In [1] other related integral transforms are also considered, one of them being so called *X-ray transformation* which maps a function to its line integrals instead of integrals over hyperplanes.

Since we are working in two dimensions, it does not make sense to differentiate with the two transformations.

In two dimensions $\boldsymbol{\theta} = (\cos\varphi, \sin\varphi)$ for some $\varphi \in [0, 2\pi]$. Let us also denote $\boldsymbol{\theta}^\perp = (-\sin\varphi, \cos\varphi)$. Now the Radon transform of f is defined as its integrals along the lines

$$L(\varphi, s) = \{\boldsymbol{x} \in \mathbb{R}^2 : \boldsymbol{x} \cdot \boldsymbol{\theta} = s\},$$

so we can write Radon transform as a function of φ and s

$$\mathcal{R}f(s, \varphi) = \int_{L(\varphi,s)} f(\boldsymbol{x}) d\boldsymbol{x}_L = \int_{-\infty}^{\infty} f(s\boldsymbol{\theta} + t\boldsymbol{\theta}^\perp) dt. \tag{2.6}$$

In the fan-beam case variables α, β are used, where β is the angle of the X-ray source, and α is the angle between an individual line and the central ray, that is the ray passing from the source through the origin, and the ray being measured. One can easily express variables φ and s using these angles

$$\varphi = \beta + \alpha \tag{2.7}$$

$$s = D\sin(\alpha) \tag{2.8}$$

where D is the distance of the source from the origin. Therefore the fan-beam transform is simply

$$Df(\alpha, \beta) = \mathcal{R}f(D\sin(\alpha), \beta + \alpha). \tag{2.9}$$

The most interesting property of Radon transform is probably the inversion formula, which shows that the function can be recovered if all the line integrals are known. The inversion formula follows from so called Fourier slice theorem, which gives a connection between Radon and Fourier transforms. The 1D Fourier transform for functions in $\mathbb{R}^2$ with respect to the first variable is defined by

$$\tilde{f}(\xi, y) = \int_{-\infty}^{\infty} f(x, y) e^{-i\xi x} dx. \tag{2.10}$$

and the 2D Fourier transform by

$$\hat{f}(\boldsymbol{\xi}) = \int f(\boldsymbol{x}) e^{-i\boldsymbol{\xi}\cdot\boldsymbol{x}} d\boldsymbol{x}. \tag{2.11}$$

Theorem 2.12. *Let $f \in L^1(\mathbb{R}^2)$, then*

$$\widetilde{\mathcal{R}f}(\xi, \varphi) = \hat{f}(\xi\boldsymbol{\theta}). \tag{2.13}$$

Proof. By definitions

$$\begin{aligned}\widetilde{\mathcal{R}f}(\xi,\varphi) &= \int_{-\infty}^{\infty} e^{-i\xi s}\mathcal{R}f(s,\varphi)ds \\ &= \int_{\mathbb{R}} e^{-i\xi s}\int_{-\infty}^{\infty} f(s\boldsymbol{\theta}+t\boldsymbol{\theta}^{\perp})dtds\end{aligned}$$

Now by change of variables $\boldsymbol{x} = s\boldsymbol{\theta} + t\boldsymbol{\theta}^{\perp}$ and $s = \boldsymbol{\theta}\cdot x$ we have

$$\widetilde{\mathcal{R}f}(\xi,\varphi) = \int_{\mathbb{R}^2} e^{-i\xi\boldsymbol{\theta}\cdot\boldsymbol{x}} f(\boldsymbol{x})d\boldsymbol{x} = \hat{f}(\xi\boldsymbol{\theta}). \tag{2.14}$$

□

Now by applying inverse Fourier transform to the Fourier slice theorem one can derive the Radon inversion formula which provides a way to reconstruct f from its line integrals.

Of course in reality all the line integrals can not be known, and thus the inversion formula can not be used in practice. Additionally we do not have accurate measurements; there is always some inherent inaccuracy of physical measurements. This noise is modeled as a normally distributed random variable $\varepsilon_{s,\theta} \sim \mathcal{N}(0,\sigma^2)$ for all θ and s. Now the measurement $m_{s,\theta}$ along the line $L = \{x \in \mathbb{R}^2 : x\cdot\theta = s\}$ is

$$m_{s,\theta} = \mathcal{R}f(s,\theta) + \varepsilon_{s,\theta}. \tag{2.15}$$

This noise causes some trouble when we consider inversion of the Radon transform, this is discussed more in detail in following sections.

2.1.1 Model discretization

For computational implementation of the reconstruction algorithms a discrete form of the measurement model is needed. In the parallel beam case the angular variable θ is sampled with J equidistant steps

$$\theta_j = \theta_1 + \left(\frac{j-1}{J}\right)\pi, \tag{2.16}$$

where θ_1 is a reference angle and $j \in \mathbb{N} :\ 1 \le j \le J$. Similarly s is also sampled with N equidistant steps

$$s_i = -S + 2\left(\frac{i-1}{N}\right)S \tag{2.17}$$

where $S > 0$ and $i \in \mathbb{N} :\ 1 \le i \le N$.

Now the measurements are taken along $k = JN$ lines L_i and we have

$$m = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_k \end{bmatrix} = \begin{bmatrix} \int_{L_1} f(x)ds_1 + \epsilon_1 \\ \int_{L_2} f(x)ds_2 + \epsilon_2 \\ \vdots \\ \int_{L_k} f(x)ds_k + \epsilon_k \end{bmatrix}. \tag{2.18}$$

To get a finite dimensional measurement model we have to pixelate the unknown f, that is divide the area into n pixels with constant attenuation coefficients in each pixel. For simplicity let us denote the vector of attenuation coefficients by $f = [f_1, f_2, ..., f_n]$. The integrals in (2.18) can now be approximated by finite sums

$$\int_{L_i} f(x)ds \approx \sum_{j=1}^{n} a_{ij} f_j. \tag{2.19}$$

The equation (2.18) now takes the form

$$m = Af + \varepsilon, \tag{2.20}$$

where $A = (a_{ij})$ is a matrix of size $k \times n$ and each entry a_{ij} the distance ith line passes through the jth pixel.

In the fan beam case the parameters α and β are sampled at regular intervals. Unlike in the parallel beam case however, the measurement are usually taken over the full circle because the measurement lines are not the same when measured from the opposite direction. The variable $\beta \in [0, 2\pi]$ is then sampled as

$$\beta_j = \beta_1 + \frac{2\pi}{J}(j-1), \tag{2.21}$$

where $j = 1, ..., J$. Similarly for α we have

$$\alpha_i = -\alpha_1 + 2\frac{i-1}{I}\alpha_1, \tag{2.22}$$

where $i = 1, ..., I$. The angle $\alpha_1 > 0$ is chosen so that f is supported in the disc of radius $r = D\sin(\alpha_1)$ centered in the origin. If we make an assumption that f is supported in the unit disk we have $\alpha_1 = \arcsin\frac{1}{D}$. Otherwise the discrete model is the same as in the parallel beam case.

2.2 Ill-posedness and regularization in CT

Above it was shown that Radon transform has an inverse transform, so it is known that a solution for the inverse problem can be found if exact projections along all the lines are known. This obviously is not the case in practical CT imaging. The data is corrupted by noise and in reality only a small part of the projection data is known. Sometimes the projection angles are limited to a smaller range than the half circle this is called *limited angle tomography*. Sometimes we only have a small amount of projections from the full range of angles, which is the case we consider. These are some of the *ill-posed* inverse problems related to CT imaging. Here we will discuss ill-posedness only briefly, for more detailed treatment of the subject see [1].

We will begin by giving a definition of well-posedness (and ill-posedness) by Jacques Hadamard.

Definition 2.23. Problem is *well-posed* if

1. Solution f exists
2. Solution f is unique
3. Solution f depends continuously on data m

problems for which at least one of the condition fails are called *ill-posed*.

In case of practical CT imaging, even the first two conditions usually fail: the measurement matrix A is often not square, and thus does not have a well defined inverse. In this case the Moore-Penrose pseudoinverse can be used. That is, the system $Af = m$ is solved in least-squares sense. Even if the unique solution exists, the last condition fails. This can be demonstrated in the discrete case by considering the singular value decomposition of the measurement matrix A. The singular value decomposition of a matrix $A \in \mathbb{R}^{m\times n}$ is

$$A = UDV^T \tag{2.24}$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices and $D \in \mathbb{R}^{m\times n}$ is a rectangular diagonal matrix. The diagonal elements of D are called the singular values of A. Now even in ideal case where $m = n$ and A is invertible we have

$$A^{-1} = VD^{-1}U^T \tag{2.25}$$

where $D^{-1} = diag\left(\frac{1}{d_1}, \ldots, \frac{1}{d_n}\right)$ and $d_1, \ldots, d_n$ are the singular values of A. Now we can write

$$A^{-1}m = f + A^{-1}\epsilon, \tag{2.26}$$

where error $A^{-1}\epsilon$ is bounded by

$$\|A^{-1}\epsilon\| \leq \|A^{-1}\|\|\epsilon\| = \frac{1}{d_k}\|\epsilon\|. \tag{2.27}$$

Now considering the ratio of relative error in the reconstruction and the relative error of the measurement we have

$$\frac{\frac{\|A^{-1}\varepsilon\|}{\|A^{-1}m\|}}{\frac{\|\varepsilon\|}{\|m\|}} = \frac{\|A^{-1}\varepsilon\|}{\|\varepsilon\|}\frac{\|m\|}{\|A^{-1}m\|} = \frac{\|A^{-1}\varepsilon\|}{\|\varepsilon\|}\frac{\|Af\|}{\|f\|} \leq \frac{\frac{1}{d_k}\|\varepsilon\|}{\|\varepsilon\|}\frac{d_1\|f\|}{\|f\|} = \frac{d_1}{d_k} \tag{2.28}$$

This is called the condition number of the matrix A. If the condition number is really large, the relative error of the reconstruction can be immensely larger than the relative error of the measurement. This applies also for the pseudoinverse solution: the pseudoinverse can be written in form

$$A^+ = VD^+U^T \tag{2.29}$$

where matrix $D^+ \in \mathbb{R}^{m\times n}$ is a diagonal matrix with entries corresponding to positive singular values d_k are $\frac{1}{d_k}$ and entries corresponding to zero singular values are zero. Now if the condition number of A is large, the minimum norm solution also amplifies noise. We shall demonstrate the ill-posedness of sparse-angle CT by computing the SVD in the results chapter.

Solving ill-posed inverse problems requires *regularization*. Regularization introduces some additional information to negate the effect of noise and missing data. The regularization strategies we will consider are formulated as a minimization of a functional of the form

$$\|Af - m\|_2^2 + \lambda R(f), \tag{2.30}$$

where $\lambda > 0$ and R is a regularization term which introduces the additional information used. This could be for example smoothness of the reconstruction, distance to some vector or sparsity with respect to some basis or frame. The regularization can then be thought as balancing between minimization of the recidual and the regularization term. The regularization parameter λ controls the effect of regularization; larger it is, more effect the regularization has on the reconstruction. In this thesis we will consider sparsity promoting regularization with respect to some orthogonal basis. In this case the functional takes the form

$$\|Af - m\|_2^2 + \lambda \sum_{k=1}^{N} |\langle f, \psi_k\rangle|, \tag{2.31}$$

where ψ_k are the basis functions.

Chapter 3

Wavelets

Where classical Fourier analysis uses simple trigonometric functions to represent more complicated functions, wavelet analysis uses *wavelets*, small waves that decay to zero at $\pm\infty$, to do the same. Wavelet transforms allow us to divide the function into components of different scales but at the same time also captures the frequency information of the function, whereas the change of frequency in Fourier transform causes changes in whole domain. Discrete versions of wavelet transforms are often used in image and signal compression, and other signal processing tasks. This chapter covers the basic properties and definitions of wavelets and wavelet transforms needed later in the thesis. We begin by considering the continuous wavelet transform, then the discrete transform in one dimension and finally two dimensional wavelet transform used in reconstruction process. This chapter is based mainly on [15] as well as [14].

3.1 The basics

We first consider continuous, one dimensional wavelet transforms. To compute the wavelet transforms one needs a family of wavelets. This is done my choosing a so called mother wavelet; a function ψ that should satisfy the property

$$\int \psi(t)dt = 0 \tag{3.1}$$

The family of wavelets is then defined by scaling and shifting the mother wavelet

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right) \tag{3.2}$$

for $a, b \in \mathbb{R}$. The parameter a is the scaling parameter, allowing wavelets to cover different frequency ranges. Usually only positive values of a are used. The parameter b on the other

hand controls the location of the wavelet. Now the continuous wavelet transform of f is defined as

$$(\mathcal{W}f)(a,b) = \int f(t)\psi_{a,b}(t)dt = \langle f, \psi_{a,b}\rangle. \tag{3.3}$$

The question immediately arising is can we know f if $\mathcal{W}f$ is known? First let us consider the Fourier transform of the wavelet $\psi_{a,b}$:

$$\hat{\psi}(\xi) = \frac{1}{\sqrt{2\pi}}\int \psi_{a,b}(t)e^{-i\xi x}dx = \frac{1}{\sqrt{2\pi}}\int |a|^{-\frac{1}{2}}\psi\left(\frac{x-b}{a}\right)e^{-i\xi x}dx. \tag{3.4}$$

Now using the substitution $t = \frac{x-b}{a}$ we have

$$a|a|^{-\frac{1}{2}}e^{-i\xi b}\frac{1}{\sqrt{2\pi}}\int \psi(t)e^{-i\xi at}dt = a|a|^{-\frac{1}{2}}e^{-i\xi b}\hat{\psi}(a\xi). \tag{3.5}$$

By using Plancherel's theorem we get a new representation for the wavelet transform

$$(\mathcal{W}f)(a,b) = \langle f, \psi_{a,b}\rangle = \frac{1}{2\pi}\langle \hat{f}, \hat{\psi}_{a,b}\rangle = \frac{a|a|^{-\frac{1}{2}}}{2\pi}\int \hat{f}(\xi)e^{ib\xi}\overline{\hat{\psi}(a\xi)}d\xi. \tag{3.6}$$

Using the representation above we can prove the resolution of identity formula. The formula shown here and it's proof are an adaptation of the Proposition 2.4.1 of [15].

Theorem 3.7 (Resolution of identity)**.** *For all $f, g \in L^2(\mathbb{R})$,*

$$\int\int \frac{1}{a^2}(\mathcal{W}f)(a,b)\overline{(\mathcal{W}g)(a,b)}dadb = C_\psi\langle f, g\rangle, \tag{3.8}$$

where

$$C_\psi = 2\pi\int |\xi|^{-1}|\hat{\psi}(\xi)|^2 d\xi. \tag{3.9}$$

Proof. By using above identities for $\mathcal{W}f$ and $\mathcal{W}g$ we have

$$\begin{aligned}&\int\int \frac{1}{a^2}(\mathcal{W}f)(a,b)\overline{(\mathcal{W}g)(a,b)}dadb\\ &= \int\int \frac{1}{a^2}\left[\frac{a|a|^{-\frac{1}{2}}}{2\pi}\int \hat{f}(\xi)e^{ib\xi}\overline{\hat{\psi}(a\xi)}d\xi\right]\left[\frac{a|a|^{-\frac{1}{2}}}{2\pi}\int \overline{\hat{g}(\xi')}e^{-ib\xi'}\hat{\psi}(a\xi')d\xi'\right]dadb.\end{aligned} \tag{3.10}$$

Now let us write

$$F(\xi) = a|a|^{-\frac{1}{2}}\overline{\hat{f}(\xi)}\hat{\psi}(a\xi) \tag{3.11}$$

and

$$G(\xi) = a|a|^{-\frac{1}{2}}\overline{\hat{g}(\xi)}\hat{\psi}(a\xi). \tag{3.12}$$

Thus (3.10) becomes

$$\int\int \frac{1}{a^2}\overline{\hat{F}(b)}\hat{G}(b)dbda = \int \frac{1}{a^2}\langle \hat{G}, \hat{F}\rangle da. \tag{3.13}$$

Again by Plancherel's theorem we have

$$\begin{aligned} &\int \frac{1}{a^2}\langle \hat{G}, \hat{F}\rangle da \\ &= \int \frac{2\pi}{a^2}\langle G, F\rangle da \\ &= 2\pi \int \frac{1}{|a|}\int \hat{f}(\xi)\overline{\hat{g}(\xi)}|\hat{\psi}(a\xi)|^2 d\xi da \\ &= 2\pi \int \hat{f}(\xi)\overline{\hat{g}(\xi)}\int \frac{1}{|a|}|\hat{\psi}(a\xi)|^2 da d\xi \end{aligned} \tag{3.14}$$

And finally using substitution $\zeta = a\xi$ in the second integral we have

$$(3.14) = C_\psi \langle f, g\rangle \tag{3.15}$$

□

The resolution of identity holds only if $C_\psi < \infty$. Using the resolution of identity we can write f as

$$f = C_\psi^{-1}\int\int \frac{1}{a^2}(\mathcal{W}f)(a,b)\psi_{a,b}dadb. \tag{3.16}$$

3.2 Discrete wavelet transform and the multiresolution analysis

Discrete versions of wavelets are defined by choosing only discrete values of a and b. The parameter a is discretized by choosing one fixed $a_0 > 1$ and taking integer powers of it. That is

$$a = a_0^m, \tag{3.17}$$

where $m \in \mathbb{Z}$. The parameter b must also depend on m, because more dilated the wavelet, the smaller translation steps are required. Let us choose $b_0 > 0$, now

$$b = nb_0a_0^m, \tag{3.18}$$

where $n \in \mathbb{Z}$. Now the discrete wavelets are

$$\psi_{m,n}(x) = a_0^{-m/2}\psi(a_0^{-m}x - nb_0). \tag{3.19}$$

Finally we have the discrete wavelet transform

$$T_{m,n}(f) = a_0^{-m/2}\int f(t)\psi(a_0^{-m}t - nb_0)dt = \langle f, \psi_{m,n}\rangle. \tag{3.20}$$

As can be seen from the regularization functional (2.31), for our purpose we are interested in mother wavelets that construct an orthonormal basis in $L^2(\mathbb{R})$. Such wavelets can be constructed using multiresolution analysis. Additionally multiresolution analysis gives us an efficient way of computing wavelet transforms. The multiresolution analysis consists of a sequence of closed subspaces of $L^2(\mathbb{R})$:

$$...V_2 \subset V_1 \subset V_0 \subset V_{-1} \subset V_{-2} \subset ... \tag{3.21}$$

so that $\bigcup V_j$ is dense in $L^2(\mathbb{R})$ and $\bigcap V_j = \{0\}$. Additionally

$$f(x) \in V_n \Leftrightarrow f(2^n x) \in V_0,\ n \in \mathbb{Z}$$
$$f(x) \in V_0 \Leftrightarrow f(x-k) \in V_0,\ k \in \mathbb{Z}$$

and there exists a *scaling function* $\phi \in V_0$, such that $\{\phi_{0,n}\}_{n\in\mathbb{Z}}$, where for all $j, n \in \mathbb{Z}$, $\phi_{j,n}(x) = 2^{-j/2}\phi(2^{-j}x - n)$, forms an orthogonal basis for V_0. One of the basic results given in [15] is that if a collection of closed subspaces satisfies requirements above, there exists an othonormal wavelet basis of $L^2(\mathbb{R})$ such that for all $f \in L^2(\mathbb{R})$,

$$P_{j-1}f = P_j f + \sum_{k\in\mathbb{Z}} \langle f, \psi_{j,k}\rangle \psi_{j,k}, \tag{3.22}$$

where P_j is the orthogonal projection onto the subspace V_j. Also the wavelet ψ can be constructed using

$$\psi = \sum_n g_n \phi_{-1,n}, \quad \text{where } g_n = (-1)^n \overline{h_{-n+1}} \quad \text{and } h_n = \langle \phi, \phi_{-1,n}\rangle. \tag{3.23}$$

Let us define the Haar multiresolution analysis and construct the corresponding Haar wavelet basis. This will be the basis used by the reconstruction algorithm later. The Haar multiresolution analysis is defined as

$$V_j = \{f \in L^2(\mathbb{R}); \forall k \in \mathbb{Z} : f|_{[2^j k, 2^j(k+1)[} = c\} \tag{3.24}$$

where c is a constant. The corresponding scaling function is

$$\phi(x) = \begin{cases} 1, & \text{if } 0 \le x < 1 \\ 0, & \text{otherwise} \end{cases} . \tag{3.25}$$

Now we have

$$h_n = \sqrt{2} \int \phi(x) \overline{\phi(2x-n)} dx = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } n = 0, 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.26}$$

and thus

$$g_n = (-1)^n \overline{h_{-n+1}} = \begin{cases} \frac{1}{\sqrt{2}} & \text{if } n = 0 \\ -\frac{1}{\sqrt{2}} & \text{if } n = 1 \\ 0 & \text{otherwise} \end{cases} . \tag{3.27}$$

This results in the Haar mother wavelet

$$\psi(x) = \begin{cases} 1 & \text{if } 0 \le x < \frac{1}{2} \\ -1 & \text{if } \frac{1}{2} \le x < 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.28}$$

One of the main results of multiresolution analysis is the Fast wavelet transform algorithm which makes computations of wavelet and inverse wavelet transforms faster and easier. Following the path given in [15] we note that based on (3.23)

$$\psi_{j,k}(x) = 2^{-j/2} \psi(2^{-j}x - l) \tag{3.29}$$

$$= 2^{-j/2} \sum_n g_n 2^{1/2} \phi(2^{-j+1}x - 2k - n) \tag{3.30}$$

$$= \sum_n g_n \phi_{j-1,2k+n}(x) \tag{3.31}$$

$$= \sum_n g_{n-2k} \phi_{j-1,n}(x), \tag{3.32}$$

it then follows that

$$\langle f, \psi_{j,k} \rangle = \sum_n \overline{g_{n-2k}} \langle f, \phi_{j-1,n} \rangle. \tag{3.33}$$

The coefficients $\langle f, \phi_{j-1,k} \rangle$ can be computed simply by using

$$\begin{aligned} \phi_{j,k}(x) &= 2^{-j/2} \phi(2^{-j}x - k) \\ &= \sum_n h_{n-2k} \phi_{j-1,n}(x), \end{aligned}$$

which means that

$$\langle f, \phi_{j,k} \rangle = \sum_n \overline{h_{n-2k}} \langle f, \phi_{j-1,n} \rangle. \tag{3.34}$$

Equations (3.33) and (3.34) give us an efficient way of computing the successive wavelet coefficients from $\langle f, \phi_{0,k} \rangle$ level by level. If we denote

$$c_k^j = \langle f, \phi_{j,k} \rangle$$
$$w_k^j = \langle f, \psi_{j,k} \rangle,$$

we can write (3.33) and (3.34)

$$w_k^j = \sum_n \overline{g_{n-2k}} c_n^{j-1} = g(-n) * c_n^{j-1} \tag{3.35}$$

and

$$c_k^j = \sum_m \overline{h_{m-2k}} c_m^{j-1} = h(-n) * c_n^{j-1}. \tag{3.36}$$

where $n = 2k, k \geq 0$ and $*$ denotes the convolution operation.

Therefore the fast wavelet transform algorithm works by successively convolving the signal c_j with low- and high pass filters $h(-n)$ and $g(-n)$, and resulting high-frequency channel w_{j+1} and the low-frequency channel c_{j+1} are downsampled by two. The process then proceeds with c_{j+1}. If the original signal has N samples, the process is continued up to $J = \log_2 N$ scales. This results in the sequence of wavelet coefficients $w_1, w_2, ..., w_J, c_J$.

From the wavelet transform above it is easy to derive an inverse transform. One upsamples the coefficients w_j, c_j by two and convolves them with synthesis filters $g(n)$ and $h(n)$, and finally sums them to get c_{j-1}. The process then continues with coefficients w_{j-1}, c_{j-1} until the signal is recovered as $f = c_0$.

3.2.1 The 2D discrete wavelet transform

Since our CT reconstructions will be two dimensional, we need two dimensional wavelet transforms. The following treatment is based on one given in [14]. The two dimensional wavelet transform is defined using four different separable functions:

$$\phi(x, y) = \phi(x)\phi(y) \tag{3.37}$$
$$\psi^H(x, y) = \psi(x)\phi(y) \tag{3.38}$$
$$\psi^V(x, y) = \phi(x)\psi(y) \tag{3.39}$$
$$\psi^D(x, y) = \psi(x)\psi(y). \tag{3.40}$$

The scaled and translated basis functions are

$$\phi_{j,m,n}(x,y) = 2^{j/2}\phi(2^j x - m, 2^j y - n)$$
$$\psi^i_{j,m,n}(x,y) = 2^{j/2}\psi^i(2^j x - m, 2^j y - n), \quad i = \{H, V, D\}$$

and we can define the discrete wavelet transform (of a discrete f) as

$$d_{j_0,m,n} = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\phi_{j0,m,n}(x,y) \tag{3.41}$$

$$c^i_{j,m,n} = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y)\psi^i_{j,m,n}(x,y), \quad i = \{H, V, D\}. \tag{3.42}$$

Inverse wavelet transform then is

$$\begin{aligned} f(x,y) =& \frac{1}{\sqrt{MN}} \sum_m \sum_n d_{j_0,m,n}\phi_{j_0,m,n}(x,y) \\ &+ \frac{1}{\sqrt{MN}} \sum_{i=H,V,D} \sum_{j=j_0}^{\infty} \sum_m \sum_n c^i_{j,m,n}\psi^i_{j,m,n}(x,y). \end{aligned}$$

The algorithm for 2D transforms can then be derived. Using the scaling function $\phi(x,y) = \phi(x)\phi(y)$ one can compute the coefficients c_{j+1} from the coefficient c_j by

$$c_{j+1,k,l} = \sum_{m,n} h_{m-2k} h_{n-2l} c_{j,m,n} \tag{3.43}$$

and three other coefficients w^H, w^V, w^D by

$$w^H_{j+1,k,l} = \sum_{m,n} h_{m-2k} g_{n-2l} c_{j,m,n}$$
$$w^V_{j+1,k,l} = \sum_{m,n} g_{m-2k} h_{n-2l} c_{j,m,n}$$
$$w^D_{j+1,k,l} = \sum_{m,n} g_{m-2k} g_{n-2l} c_{j,m,n}.$$

So each coefficient in the next scale is computed by 2D-convolving c_j with the separable filters and downsampling both dimensions by factor 2.

The recovery of the image from the coefficients is analogous to previously introduced 1D case, that is upsampling by 2 and 2D-convolving with the synthesis filters.

Chapter 4

Iterative thresholding algorithm and automatic parameter selection

In this chapter we address the iterative soft thresholding algorithm (ISTA) as sparsity promoting reconstruction algorithm for CT imaging. Additionally we will discuss the selection of the regularization parameter using simple feedback controllers.

4.1 Iterative soft-thresholding algorithm for linear inverse problems

The reconstruction algorithm studied in this thesis involves regularization in *Besov spaces*. The main observation used is that wavelet bases in L^2 are unconditional bases in Besov spaces, giving us a way to compute norms used in regularization. The problem is solved using an iterative thresholding algorithm introduced in [2] resulting in a sparse solution.

The general form of the functional being minimized is

$$\Phi_{w,p}(f) = \|Af - m\|^2 + \sum w_{i,j} |\langle f, \psi_{i,j} \rangle|^p, \tag{4.1}$$

where $w = (w_{i.j})$ is a sequence of strictly positive weights and $(\psi_{i,j})$ is an orthogonal basis. In our special case $p = 1$, $w_{i,j} = \mu$ for all $i, j \in \mathbb{Z}$ and $\mu \in \mathbb{R}$. Also the othogonal basis used is some wavelet basis. Now we can write (4.1) as

$$\Phi_{\mu}(f) = \|Af - m\|^2 + \mu \sum |\langle f, \psi_{i,j} \rangle| \tag{4.2}$$

The minimization algorithm is derived using a sequence of so called surrogate functionals which are easier to minimize and for which the minimizers have the minimizer of (4.2) as a limit.

The surrogate functional used is of the form

$$\Phi_{\mathrm{SUR}}(f;a) = \Phi_\mu(f) - \|Af - Aa\|^2 + \|f - a\|^2 \tag{4.3}$$

for some a in the same space as f, in our case a Besov space. The point of the iterative algorithm is to choose some initial f^0, and then

$$f^n = \operatorname{argmin}\left(\Phi_{\mathrm{SUR}}(f; f^{n-1})\right) \tag{4.4}$$

for each $n = 1, 2, \ldots$. In [2] the minimizer of the surrogate functional (4.3) for arbitrary a was proven to be

$$f = \mathbf{S}_\mu(a + A^*(m - Aa)), \tag{4.5}$$

and thus we have iterations

$$f^n = \mathbf{S}_\mu(f^{n-1} + A^*(m - Af^{n-1})) \tag{4.6}$$

where $\mathbf{S}_\mu(g) = \sum S_\mu(\langle g, \psi_{i,j}\rangle)\psi_{i,j}$ with a soft-thresholding function

$$S_\mu(x) = \begin{cases} x + \frac{\mu}{2} & \text{if } x \leq -\frac{\mu}{2} \\ 0 & \text{if } |x| < \frac{\mu}{2} \\ x - \frac{\mu}{2} & \text{if } x \geq \frac{\mu}{2} \end{cases}. \tag{4.7}$$

So the operator $\mathbf{S}_\mu$ first computes the wavelet transform of g, applies soft thresholding to set small wavelet coefficients zero and then computes the inverse wavelet transform. Thus these iterations promote sparsity with respect to the wavelet basis used.

In CT we of course know that our target f is a non-negative function, since we assume that no material amplifies the beam. Thus we also force every iterate to be non-negative by setting

$$f^n(x) = \max\{0, \mathbf{S}_\mu(f^{n-1} + A^*(m - Af^{n-1})(x)\}. \tag{4.8}$$

It was proven in [2] that the sequence (4.4) converges to the minimizer of (4.2), it is unclear however how forcing non-negativity affects this convergence. This problem could be avoided by using the primal-dual fixed point (PDFP) algorithm introduced in [16] instead.

4.2 The choice of regularization parameter

We use a sparsity based parameter selection method closely related to methods introduced in [6] and [7] for selecting the thresholding parameter μ. These methods assume a priori known level of sparsity in the wavelet domain and thus one has to choose μ so that the resulting

reconstruction has approximately the same level of sparsity. The simplest version of this kind of sparsity based method [6] involves computing the reconstructions for multiple values of μ and choosing a one corresponding to approximately the desired level of sparsity, whereas the method in [7] fits an interpolation curve to the pairs $\{\mu_i, S(\mu_i)\}$ where $S(\mu_i)$ is the number of non-zero wavelet coefficients for μ_i.

These kinds of methods are based on the observation that when μ is large, all the wavelet coefficients are set to zero, and thus the sparsity is at its maximum, on the other hand when parameter tends to zero there is less and less regularization and thus the result is more erratic and sparsity is at its minimum. In between these extremes there seems to be more or less monotonous change in sparsity.

The method we consider dynamically changes μ on every iteration of reconstruction so that the desired level of sparsity is reached. This is achieved using a relatively simple feedback control system. It is worth mentioning that the algorithm assumes arbitrary starting point f_0 for the iteration [2] , and thus one could think the process of varying μ as beginning of a new iteration with a better initial f_0.

4.2.1 PID controllers

The methods for selecting the thresholding parameter used in this thesis are related to proportional-integral-derivative (PID) controllers. PID controller compares the current ratio of nonzero coefficients to the desired level and changes μ accordingly. Here the algorithm is considered as a *plant* which outputs the ratio of nonzero coefficients on each iteration, and takes μ as an input which should be varied so that desired sparsity is reached. Let $y(t)$ be the ratio of nonzero coefficients at time t, and y_{prior} be the desired ratio of nonzero coefficients. Now we define the error e by the difference of the two ratios $e = y(\mu, t) - y_{prior}$. Notice that the error can be negative when μ is too large, and is positive when μ is too small. This error is fed to the PID controller which determines how μ should be changed to correct some of the error. The PID control for μ at time t_k is defined as

$$\mu(t) = K_p e(t) + K_i \int_0^{t_k} e(t)dt + K_d e'(t) \tag{4.9}$$

where $K_p, K_i, K_d \geq 0$. The first term is called the proportional term of the controller. It changes μ proportional to the error. The second term is the integral term. It takes into account the magnitude of error and its duration: it takes the past errors into account, and helps with the residual errors of the proportional term. The last term is the derivative term which takes the slope of the error into account: if the slope is steep and we are still far away from optimal μ, less change in μ is probably required. Derivative term thus slows down the controller.

The discretization of the PID controller can be done in a few different ways. The most obvious one is the *positional form* of the controller. This form is derived simply by discretizing

the integral as a sum, and the derivative as a difference. In our case the positional form seems unintuitive, because the proportional part of the controller does not seem to make any sense: if the error is negative, the proportional part of the controller tries to set the μ negative. Additionally the positional form is affected by *integral windup*, which refers to the case where the integral term accumulates the error so much that it causes a large overshoot or undershoot to decrease the integral of the error. This can be a problem when μ reaches the value where $y(t) = 0$ or $y(t) = 1$. Obviously in this case increasing (or decreasing) μ further does not help, but the controller could still continue increasing it. This is comparable to a situation in industrial systems where a controller tries to open a valve more than 100%. In this case the error can not accumulate fast enough to counteract the previous positive or negative summation. Therefore μ keeps increasing or decreasing while the error stays the same and the control is lost. Similar problems appear if system has other significant nonlinearities besides the one mentioned above, for example if at time t varying μ in an interval $[\mu_n, \mu_m]$ does not change the output y but the error persists.

We use the *velocity form* of the controller, which computes the change of μ instead. We now derive the velocity form from (4.9). First let us take the time derivative of (4.9)

$$\frac{d\mu(t)}{dt} = K_p\frac{de(t)}{dt} + K_i e(t) + K_d\frac{d^2e}{dt^2}. \tag{4.10}$$

Now by replacing derivatives with backward differentiation we get

$$\frac{\mu(t_k) - \mu(t_{k-1})}{h} = K_p\frac{e(t_k) - e(t_{k-1})}{h} + K_i e(t_k) + K_d\frac{e(t_k) - 2e(t_{k-1}) + e(t_{k-2})}{h^2}.$$

Solving for $\mu(t_k)$ results in

$$\begin{aligned}\mu(t_k) =&\mu(t_{k-1}) + hK_i e(t_k) + K_p[e(t_k) - e(t_{k-1}] \\ &+ \frac{K_d}{h}[e(t_k) - 2e(t_{k-1}) + e(t_{k-2}]\end{aligned}$$

Since we use a unit timestep, the above equation can be written in form

$$\mu_k = \mu_{k-1} + K_i e_k + K_p[e_k - e_{k-1}] + K_d[e_k - 2e_{k-1} + e_{k-2}]. \tag{4.11}$$

We also write the above equation in a form

$$\mu_k = \mu_{k-1} + \Delta\mu, \tag{4.12}$$

where

$$\Delta\mu = K_i e_k + K_p[e_k - e_{k-1}] + K_d[e_k - 2e_{k-1} + e_{k-2}]. \tag{4.13}$$

So using the velocity form of the controller the threshold is updated incrementally, letting the integral control drive the controller. It is still possible to get negative thresholds using the controller, especially if K_p, K_i and K_d are chosen poorly. We deal with this by replacing (4.12) by

$$\mu_k = \max(0, \mu_{k-1} + \Delta\mu). \tag{4.14}$$

Let us take a closer look of individual parts of the velocity form of the controller. The first term in (4.13) is the driving force of the velocity form: it drives μ to correct direction, increasing it when it is too small and decreasing it when it is too large. Let us assume that $K_i, K_d = 0$, now the controller has the form

$$\mu_{k+1} = \mu_k + K_i e_k \tag{4.15}$$

where $1 \geq \alpha \geq 0$. Now if μ_n is too small, e is positive and thus $\mu_{k+1} > \mu_k$. On the next iteration one then expects the ratio of nonzero coefficients to decrease. On the other hand if μ_k is too large, e_k is negative and thus $\mu_{n+1} < \mu_n$ and ratio of nonzero coefficients should increase on next iteration. If K_i is too large, one would expect the value of μ change too much, especially when the difference e_k is small: the correct sparsity is not reached. On the other hand if K_i is too small, reaching desired sparsity takes longer and longer. This thinking suggests that K_i could be chosen by taking a large initial value and decreasing it as the error decreases, thus hopefully stabilizing the system. This thinking is the basic premise of the adaptive tuning method discussed later. With the velocity form integral windup is easier to avoid, as we can avoid changing μ when the system output y reaches its limits.

The next term of (4.12) corresponds to the proportional part of (4.9). In the velocity form the proportional part works as a sort of velocity control, taking the slope of the error into account. If we again assume that $K_i, K_d = 0$ the controller is of the form

$$\mu_k = \mu_{k-1} + K_p(e_k - e_{k-1}). \tag{4.16}$$

Now if the difference between errors is large, the change in μ will be large. On the other hand if the difference between errors is close to zero, even if the error still persists, the change in μ is small. This causes the so called *steady-state error* which is typical for proportional controllers. This suggests that if we apply control only after the algorithm has converged, the proportional control would not change anything. Pure proportional control is thus undesirable in our case. When used in unison with the integral control, proportional control makes much more sense: the integral control removes the steady-state error while proportional control lets us control the slope of the path. If the error e_k is positive, that is μ is too small, and the derivative of the error is negative, the integral control will then try to increase μ while the proportional control tries to decrease it. This means that the proportional control would slow down the convergence of

the method, but at the same time there is hope that the controller would not cross the desired level of sparsity often. On the other hand we could try to use negative coefficient K_p, which would make the convergence faster, but would probably make the system more unstable. The velocity formulation shows one of the drawbacks of the proportional control; because it involves differentiation some noise amplification should be expected.

The derivative term of (4.9) does not make a lot of sense when used alone. The point of the derivative controller is to predict future behavior of the system and thus provide stability. From the velocity form (4.12) one can see that it corresponds to the second derivative of the error signal, thus it controls the concavity of the path of the process. When the error curve is concave, the second derivative is positive and μ is increased thus driving the curve towards convexity. On the other hand when the curve is convex, the second derivative is negative and μ is then decreased. This process could give stability to the proportional part of the controller. In practice the derivative control can however increase instability: derivatives can amplify noise or high frequency measurements. This can be avoided for example by filtering out high frequency measurements. See [17] for detailed analysis and possible solutions to the problem. If this noise amplification can be minimized, the derivative term could be tuned to combat the instability introduced by proportional and the integral terms.

To decrease the effect of noise in the proportional and derivative terms, the process variable, that is the level of sparsity at previous iteration, is put through a simple *low-pass filter* and filtered output is then fed to the proportional and derivative terms of the controller. The low pass filter used is of the form

$$F(s_k) = (1 - r)F(s_{k-1}) + rs_k \tag{4.17}$$

where $0 < r \leq 1$ is the smoothing factor. Generally the amount of smoothing used should be dependent on the parameters used, however to simplify things in our case the smoothing factor is a constant. Integral term does not need the filtering, as it is noise robust already. Filtering can slow down the integral control even further thus causing accumulation of the integral and thus oscillative behaviour. With these considerations the system can be summarized in a diagram form 4.1. Other kinds of filters could be considered for the job, and one might for example want to use different smoothing factors r for proportional and derivative terms. The PID controlled iterative soft thresholding algorithm is summarized in algorithm 1.

4.2.2 Remarks on tuning of PID controllers

The tuning of a PID controller refers to the process of finding values of parameters K_p, K_i and K_d which produce the desired effect on the process being controlled. That is, trying to stabilize the process variable close enough to the set point. Tuning has an effect on the speed of convergence, over- or undershoot and tracking properties of the controller.

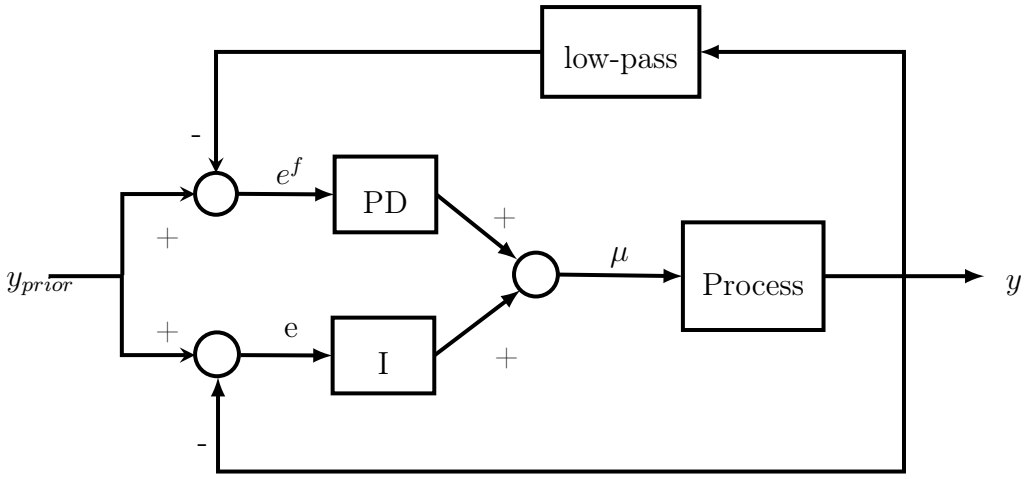


Figure 4.1: The diagram of the system.Process is the system (4.20) and the low pass filter is the shown in equation (4.17). e^f is the error computed using the output of the low-pass filter.

Algorithm 1 PID controlled ISTA

Inputs:

y_{prior}	the desired ratio of nonzero wavelet coefficients
K_p, K_i, K_d	the PID gains
m	the measured data
A	the measurement matrix
μ_0	the initial thresholding parameter
I	the maximum number of iterations
ϵ_1, ϵ_2	the tolerances for the stopping rule

$\mathbf{f}^0 = \mathbf{0},\ i = 0,\ e = 1 - y_{prior}$
while $|e| > \epsilon_1$ or $\|f^k - f^{k-1}\| > \epsilon_2$ **do**
  $f^i = \mathcal{W}^{-1} S_\mu \mathcal{W}(f^{i-1} + A^*(m - Af^{i-1}))$
  $y^i = \#\{c : c \in \mathcal{W}f^i,\ c > 0\}/\#(\mathcal{W}f^i)$
  $e_f^i = F(y_i) - y_{prior}$
  $e^i = y^i - y_{prior}$
  $\mu_k = \max(0, \mu_{i-1} + K_i e_i + K_p(e_f^i - e_f^{i-1}) + K_d(e_f^i - 2e_f^{i-1} + e_f^{i-2}))$
  $i = i + 1$
end while

While PID controllers can be tuned manually, it requires some expertise on the subject and is probably only possible in small scale. Many automatic approaches for tuning exist, most of which use simple tests to fit a model of some kind to the process being controlled. Some methods of tuning can be applied *online*, where the tuning parameters are changed during the process control, while others are applied *offline*. Most well known tuning method is the Ziegler-Nichols method [10] which has variations for P, PI, PD and PID control. When dealing with the sparsity promoting regularization, the situation and the priorities for the controller are somewhat different to most industrial processes. First of all we do not have any delicate parts which could break if controller causes oscillations or if the control is applied too quickly. Additionally the main goal is not to stabilize the process indefinitely, but to get good reconstructions as fast as possible. On the other hand in the case of iterative soft thresholding algorithm the process being controlled is dependent on the data used and it is unclear how process dynamics are affected by the data. For this reason one might want to focus on tuning strategies which involve as few assumptions on the process model as possible. These could involve, for example, heuristics and adaptive tuning methods.

In this thesis we only consider the adaptive integral control, as well as the neural network variant of the PID given in the following subsection. In the case of pure integral control (4.15) some simple adaptive tuning strategies exist. One such method was introduced in [11], of which the following method is a variation of [18]

$$K_i = e^{p_k \left(\int e(t) dt \right)^2} - 1 \tag{4.18}$$

where

$$p_i = p_0 e^{-\int e(t)^4 dt}. \tag{4.19}$$

When the error increases the value of K_i increases to try to force the integral of error to zero, on the other hand the p_i decays over time making K_i smaller and thus hopefully stabilizing the system. The value p_0 has to be chosen properly for the method to work. In our case we choose p_0 by computing the wavelet coefficients of a backprojection reconstruction, sorting them by absolute value and choosing the median of the $(1 - S_d) * 100\%$ smallest coefficients by absolute value. The median was used instead of the mean to make sure the value chosen is small enough. The implementation of the method is summarized in the algorithm 2.

It is still an open question whether the algorithm in combination with I, PI, or PID controllers actually converges. In a linear case properly tuned feedback controllers are guaranteed to converge to the desired value if the system being controlled is stable enough. In a nonlinear case, however, the system can have different behaviour in different operating regions and thus the controller could work well in one region but poorly in another. The system in our case is nonlinear. The iterative soft thresholding algorithm can be expressed as a discrete time open

loop system

$$\begin{cases} x_{i+1} = f_i + A^T(m - Af_i) \\ f_{i+1} = \mathbf{S}_\mu(x_{i+1}) \\ y_{i+1} = \frac{\#\{w : w \in W,\, |v| > 0}{\#W} \end{cases} \tag{4.20}$$

where W is the collection of wavelet coefficients of f_{i+1} and $\#S$ is the number of elements in set S. To prove the convergence of the system controlled using the adaptive integral control one would have to prove the integral controllability of the above system, that is show that there exists a $k^* > 0$ so that for all $K_i \in (0, k^*)$ the system with integral control is stable and asymptotically tracks the desired level of sparsity. In this thesis we assume that the system is close to linear when $\mu \in [0, \mu_{max}]$, where μ_{max} corresponds to the smallest threshold which results in $y_i = 0$ as $i \to \infty$, and therefore standard PID control is used.

Algorithm 2 Integral controlled ISTA with adaptive tuning

Inputs:

y_{prior}	the desired ratio of nonzero wavelet coefficients
m	the measured data
A	the measurement matrix
I	the maximum number of iterations
ϵ_1, ϵ_2	the tolerances for the stopping rule

$p_0 = \text{median}(\text{Sort}(\mathcal{W}A^*m))$
$\mathbf{f}^0 = \mathbf{0},\ i = 0,\ e = 1 - y_{prior}$
while $|e| > \epsilon_1$ or $\|f^k - f^{k-1}\| > \epsilon_2$ **do**
  $f^i = \mathcal{W}^{-1}S_\mu\mathcal{W}(f^{i-1} + A^*(m - Af^{i-1}))$
  $y_i = \#\{c : c \in \mathcal{W}f^i,\ c > 0\}/\#(\mathcal{W}f^i)$
  $e_i = y_i - y_{prior}$
  $p_i = p_0 e^{-\sum_{t=0}^{i} e_t^4}$
  $K_i = e^{p_i(\sum_{t=0}^{k} e_t)^2} - 1$
  $\mu_k = \max(0, \mu_{i-1} + Ke_i)$
  $i = i + 1$
end while

4.2.3 PIDNN

As an example of an adaptive controller we will briefly discuss a variation on the classic PID controller called the PID neural network (PIDNN) [12]. The idea of this adaptive method is to

combine the simplicity of the PID controller with the strength of the backpropagation learning algorithm. Other similar approaches use neural networks for approximating the process, but here the controller is implemented as a neural network. The neural network then learns to control the process using a simple backpropagation algorithm. The stucture of the PIDNN can be seen in diagram 4.2. The nodes of the network have three different sorts of transfer functions. The input neurons (1 and 2), the P neuron and the output neuron (O) have a simple piecewise linear transfer function

$$P(x) = \begin{cases} 1, & x > 1 \\ x, & -1 < x < 1 \\ -1, & x < -1 \end{cases} \tag{4.21}$$

The I neuron has a summing transfer function

$$I(x_k) = \begin{cases} 1, & I(x_{k-1}) + x_k > 1 \\ I(x_{k-1}) + x_k & -1 < I(x_{k-1}) + x_k < 1 \\ -1 & I(x_{k-1}) + x_k < -1 \end{cases} \tag{4.22}$$

and finally the D neuron has a differential transfer function

$$D(x_k) = \begin{cases} 1, & x_k - x_{k-1} > 1 \\ x_k - x_{k-1}, & -1 < x_k - x_{k-1} < 1 \\ -1, & x_k - x_{k-1} < -1 \end{cases} \tag{4.23}$$

Inputs for each hidden layer (P, I and D neurons) node is a weighted sum of the outputs of the input layer. The weights are denoted by $w_{i,j}$, where $i = 1, 2$ indexes the output nodes and $j = P, I, D$ indexes hidden nodes. The input for the output neuron is the weighted sum of output of the hidden layer. The weights are denoted as w_i, where $i = P, I, D$ indexes the hidden layer nodes.

Now if we set the weights as $w_{1,j} = -1$,$w_{2,j} = 1$ and $w_P = K_p$, $w_I = K_i$ and $w_D = K_d$ the output of the network is equal to the output of a standard PID controller (4.9). Our goal then is to use a backpropagation algorithm to learn the weights. The functional being minimized is

$$J = \frac{1}{N} \sum_{k=1}^{N} e_k^2. \tag{4.24}$$

Now by following the standard gradient method the weights from the hidden layer to the output layer at iteration n are

$$w_i^n = w_i^{n-1} - \eta \frac{\partial J}{\partial w_i}, \quad i = P, I, D \tag{4.25}$$

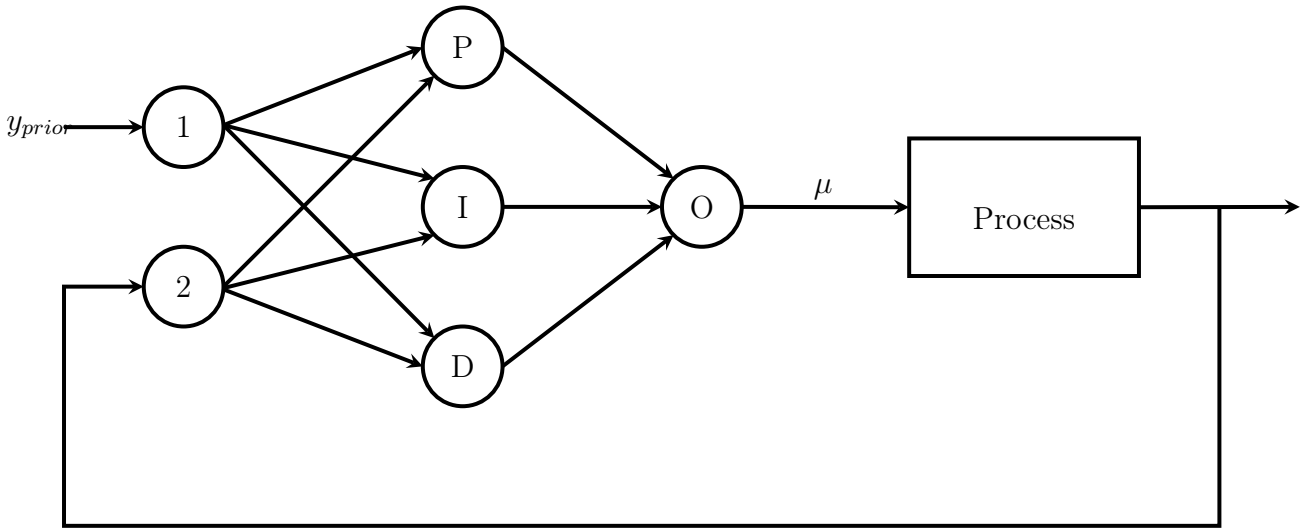


Figure 4.2: The diagram of the PIDNN. The process is the system (4.20)

where $\eta < 1$ is the learning rate and

$$\frac{\partial J}{\partial w_i} = -\frac{2}{n-1}\sum_{k=1}^{n-1} e_k \frac{e_k - e_{k-1}}{\mu_k - \mu_{k-1}} x_j^k = -\sum_{k=1}^{n-1} \delta_j^k x_j^k \tag{4.26}$$

where x_j^k is the output of the hidden node $j = P, I, D$. Similarly one can compute the corresbonding partial derivatives for weights from input layer to the hidden layer:

$$\frac{\partial J}{\partial w_{i,j}} = -\sum_{k=1}^{n-1} \delta_j^k \frac{x_j^k - x_j^{k-1}}{u_j^k - u_j^{k-1}} x_i^k = -\sum_{k=1}^{n-1} \delta_i^k x_i^k \tag{4.27}$$

where u_j^k is the input of an hidden layer neuron, x_i^k is the output of an input layer neuron. And now the input to hidden layer weights are updated by

$$w_{j,i}^n = w_{j,i}^n - \eta \frac{\partial J}{\partial w_j}, \quad j = 1, 2 \text{ and } i = P, I, D \tag{4.28}$$

Now to use the PIDNN, one only has to initialize the network weights using for example the standard PID-initalization and choose the learning rate. Effectively the three tuning parameters are therefore replaced by the learning rate. The algorithm is shown in 3.

One might expect a number of problems with such a network which make this method probably suboptimal. First of all learning takes time, and therefore makes computations more

expensive. Since the PIDNN is not in an incremental form (4.12) it is vulnerable to integral windup, especially because the learning process could be a cause for additional disturbances. Additionally implementing filters for the inputs and outputs of individual nodes is harder. On the other hand the network should be capable of adapting to a number of situations, like changing setpoints. Overall the PIDNN might be more useful in cases where the main importance is to keep the process variable as close to the setpoint as possible for long periods of time, such as many industrial processes are. This is not the case with our sparsity based regularization parameter selection, as the main goal is to get a good reconstruction preferably as fast as possible.

Algorithm 3 PIDNN controlled ISTA

Inputs:

y_{prior}	the desired ratio of nonzero wavelet coefficients
η	the learning rate
W	a 3×2 matrix containing the initial input to hidden layer weights
$\mathbf{w}$	initial hidden to output layer weights
m	the measured data
A	the measurement matrix
I	the maximum number of iterations
ϵ_1, ϵ_2	the tolerances for the stopping rule

$\mathbf{f}^0 = \mathbf{0},\ i = 0,\ e = 1 - y_{prior}$
while $|e| > \epsilon_1$ and $\|f^k - f^{k-1}\| > \epsilon_2$ **do**
  $\mathbf{y} = [P(y_{prior}), P(y)]$
  $u = W\mathbf{y}^T$
  $v = [P(u_1), I(u_2), D(u_3)]$
  $\mu = P(w \cdot v)$
  $f^i = \mathcal{W}^{-1} S_\mu \mathcal{W}(f^{i-1} + A^*(m - Af^{i-1}))$
  $y_i = \#\{c : c \in \mathcal{W}f^i,\ c > 0\}/\#(\mathcal{W}f^i)$
  update the network weights according to (4.25) and (4.28)
  $i = i + 1$
end while

Chapter 5

Materials and methods

This chapter is dedicated to materials and methods used to asses the effectiveness of our algorithms. First we will introduce the datasets used, and then some measures used to asses the effectiveness of different parameters of the PID controllers.

5.1 Simulated and real X-ray data

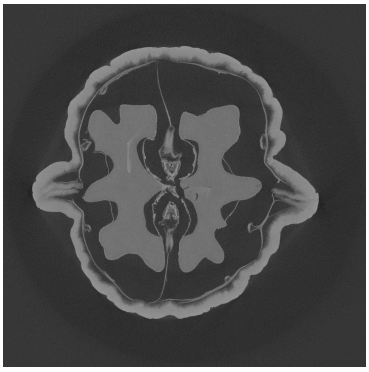
The different versions of the algorithm were tested using a simulated dataset and a real X-ray dataset. The phantom used for the simulated dataset was a standard Shepp-Logan phantom 5.1a of resolution 328×328. The projection data was computed using the parallel beam radon transform provided by Matlab with 30 uniformly distributed measurement angles. Finally some Gaussian noise was added ($\sigma = 0.1$). The desired ratio of nonzero wavelet coefficients was determined by determining the ratio of nonzero wavelet coefficients of the ground truth ($y_{prior} \approx 0.045$).

The walnut dataset [19] used consists of fan beam projection data of resolution 328×328 from 30 uniformly distributed projection angles. The high resolution ground truth image of the walnut computed with the filtered backprojection algorithm using 360 projection angles can be seen in figure 5.1b. The desired ratio of nonzero wavelet coefficients used was $y_{prior} = 0.6$, obtained by

5.2 Tools for analyzing the results

Our main motivation is to show that the PID control approach can work as a way to choose the regularization parameter, secondly we would like to show how varying the parameters of the controller affect the reconstruction process. This is mainly done using figures showing the

(a) the Shepp-Logan phantom



(b) walnut

Figure 5.1: The ground truth images from 360 projection angles

evolution of the ratio of nonzero coefficients as the iteration progresses. However, sometimes the figures do not paint the whole picture. For this reason we will use a few additional measures.

Overshoot refers to the transient state of the system where the system output exceeds the steady state, similarly undershoot refers to the case where transitory outputs are lower than the steady state value. In our case we will expect a large undershoot, because the algorithms are initialized at $\mu = 0$. To measure the undershoot we shall define *percentage undershoot*. Let y_u be the minimum value of the output in the undershooting state. Now the percentage undershoot is just the relative difference of the minimum value and the desired sparsity level:

$$PO = \frac{|y_u - y_{prior}|}{y_{prior}}. \tag{5.1}$$

We will also measure the time it takes to recover from the undershoot, that is the number of iterations between the first time the level of sparsity dips below the desired level and the iteration when it either returns back above the desired ratio or the process is stopped. Additionally we will measure the negative integral absolute error (NIAE) in the undershooting state.

$$NIAE = \int_{t_0}^{t_1} |y(t) - y_{prior}|dt \tag{5.2}$$

where t_0 is the time when undershoot begins, and t_1 is the time when the system recovers from the undershoot. Generally smaller these values are, better the controller performs.

The results will be compared to two classical reconstruction methods: the filtered back-projection and Tikhonov regularization. Classical Tikhonov regularization is formulated as a minimization problem

$$\underset{f}{\operatorname{argmin}}\{\|Af - m\|_2^2 + \alpha\|f\|_2^2\}, \tag{5.3}$$

where $\alpha > 0$ is the regularization parameter. The implicit solution of (5.3) is

$$\left(A^T A + \alpha I\right) f = A^T m, \tag{5.4}$$

from which the explicit solution can easily be computed. However, for a large scale problem such as the tomography in high resolutions, inverting the matrix $A^T A + \alpha I$ is not viable. We will solve this by applying the conjugate gradient method [20, p. 7] to the implicit solution (5.4).

The regularization parameter α is chosen using the L-curve method [21], that is Tikhonov reconstructions using a range of parameters $0 < \alpha_1 < \alpha_2 < \ldots < \alpha_P$ are computed and the points $(\|Af - m\|_2, \|f\|_2)$ are plotted in log-log scale. The α corresponding to the corner of the L-shaped curve is selected. The corner is detected by finding the point of maximum curvature. The basic idea of the L-curve method is to find the balance between the two norms.

Chapter 6

Results

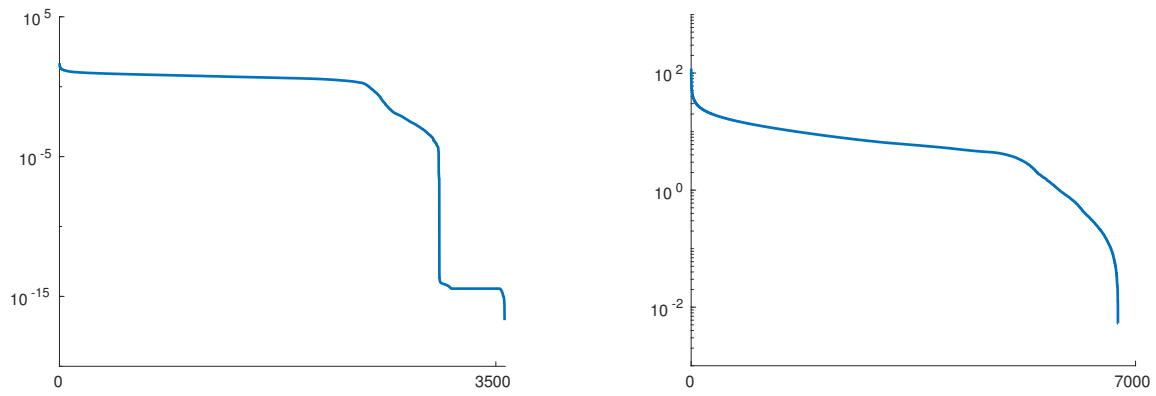
In this chapter we demonstrate the method with experiments. But first we demonstrate the ill-posedness of the problem by taking a look at the singular value decomposition of the measurement matrix. In all of the cases considered we use the system 4.1 with a filter (4.17) where $r = 0.1$. The default parameters for the stopping rule were $\epsilon_1 = \epsilon_2 = 0.001$. First we will change the tuning parameters manually, showing the effects of varying each parameter. Then the adaptive integral control (4.18) will be considered. Finally we will show the results obtained from the PIDNN controller.

6.1 The SVD of the measurement matrix

We shall demonstrate the ill-posedness of the problem by examining the singular value decompositions of the measurement matrices in the parallel beam case. The SVDs of matrices corresbonding to reconstructions of sizes 82×82 using 180 and 30 projection directions were computed. The plots of the singular values in logarithmic scales can be seen in the figure 6.1. Corresponding condition numbers are $c_{180} = 2.282 \cdot 10^4$ and $c_{30} = 8.977 \cdot 10^{31}$. This shows that fewer measurement angles result in a more ill-posed inverse problem.

6.2 The effects of tuning parameters

Here we consider the effects of varying different tuning parameters of the PID controller. First the I parameter is considered separately. Then we will consider PI and PID controllers where P and D parameters are varied.



(a) (b)

Figure 6.1: Singular values of measurement matrices corresponding to (a) 30 measurement angles (b) 180 measurement angles

Integral control

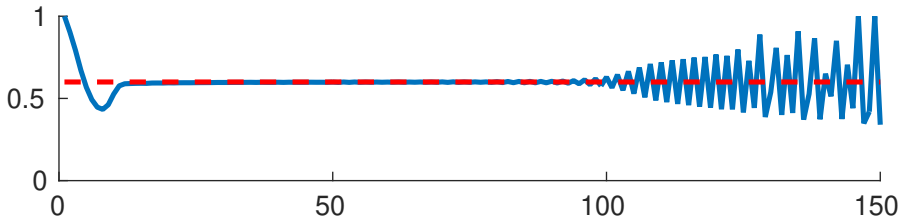
Let us see the effect of integral control (4.15). The results with three different values of K_i were computed. The evolutions of the ratio of nonzero wavelet coefficients for different values of K_i can be seen in the figure 6.2. To produce these images, some variations of the stopping rule had to be used. For the oscillating case, the iteration was stopped manually at 150 iterations. For other two cases the standard stopping rule was used.

PI control

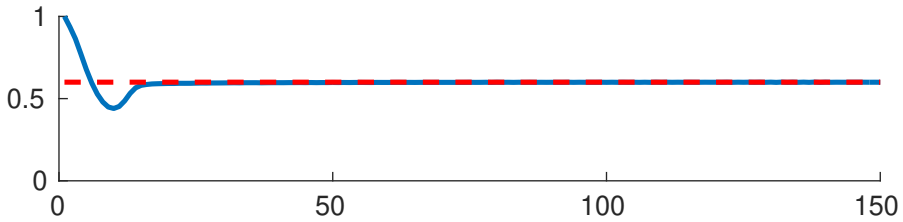
Next let us demonstrate the PI-control. Let us fix $K_i = 0.0001$ and vary
$K_p = 0.00001, 0.0001, 0.0005$. The results can be seen in figure 6.3. All three were achieved in 114 iterations. The reconstructions were virtually identical: the relative $L2$ differences between the three reconstructions were under 1% in all cases. The table 6.1 shows the change in overshoot as K_p is increased.

PID: varying D

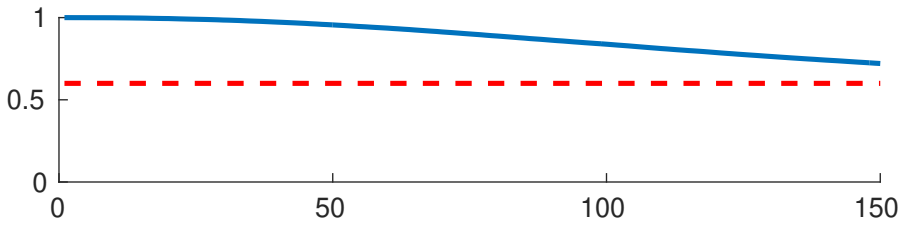
The P and I parameters were fixed $K_i = K_p = 0.0001$ and K_d parameter was then varied. All results were achieved in 114 iterations. The relative $L2$-norm differences between reconstructions were below 1%. The figure 6.4 shows the ratio of nonzero coefficients as the iteration progresses. The table 6.2 shows how K_d affects the undershooting.



(a)



(b)



(c)

Figure 6.2: The effect of varying the integral parameter K_i in the case of the walnut data. a) $K_i = 0.0003$ too large, oscillation appears b) $K_i = 0.0002$ just right c) $K_i = 0.0000001$ too small; reaching desired level takes a long time

Table 6.1: Change of the undershoot as K_p is increased

K_p	PO	TU	NIAE
0	0.2572	105	1.7147
0.0001	0.2062	104	1.3048
0.0002	0.1592	104	0.9454
0.0003	0.1136	11	0.3922
0.0004	0.0721	6	0.1656
0.0005	0.0319	4	0.0485

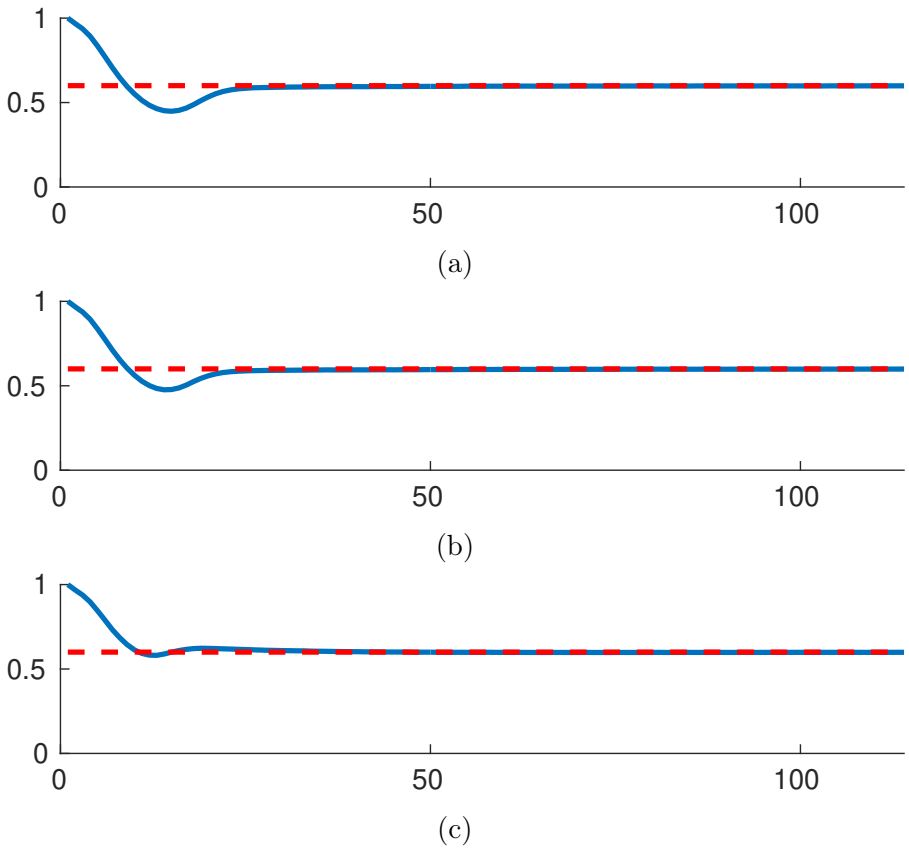


Figure 6.3: The effect of varying the proportional parameter K_p with a fixed K_i in the case of the walnut data. a) $K_p = 0.00001$ b) $K_p = 0.0001$ c) $K_p = 0.0005$

Table 6.2: Change of the undershoot as K_d is increased

K_d	PO	TU	NIAE
0	0.2062	104	1.3048
$2 \cdot 10^{-5}$	0.2047	105	1.3353
$4 \cdot 10^{-5}$	0.2028	105	1.3293
$6 \cdot 10^{-5}$	0.2026	105	1.3220
$8 \cdot 10^{-5}$	0.2039	105	1.3644
0.0001	0.2375	107	1.5743
0.0005	0.7224	112	5.0075

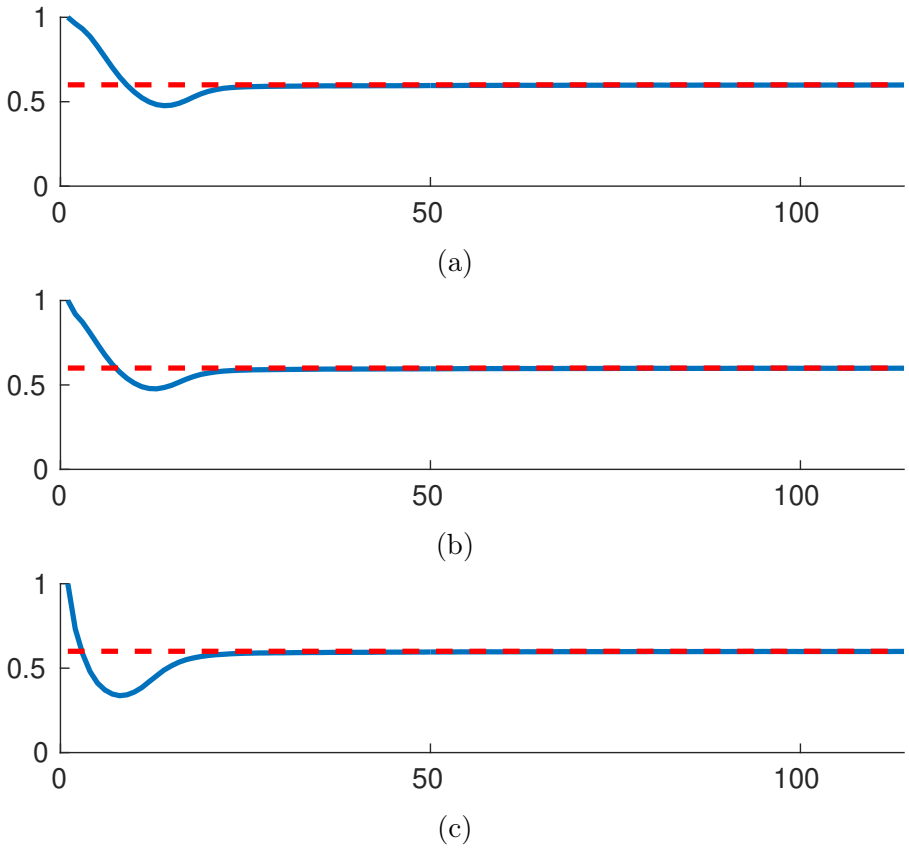
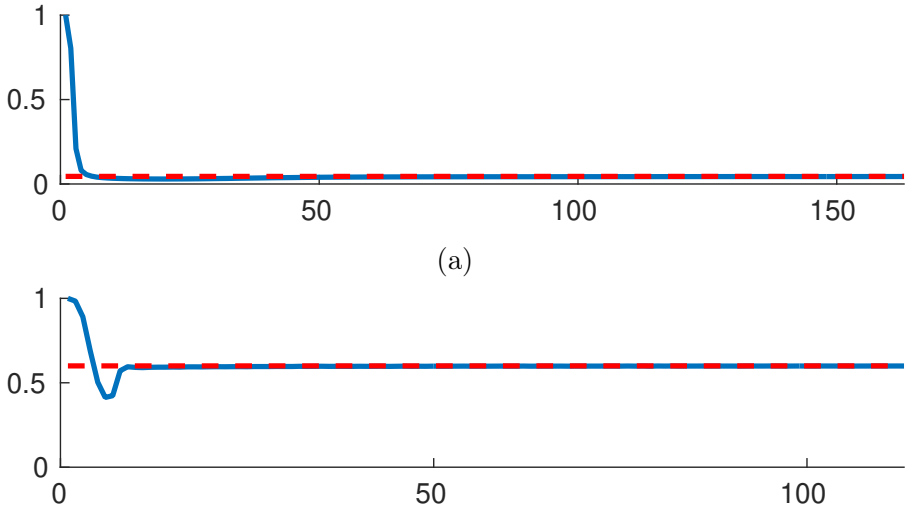


Figure 6.4: The effect of varying the derivative parameter K_d with a fixed K_i, K_p in the case of the walnut data. a) $K_d = 0.00001$ b) $K_d = 0.0001$ c) $K_d = 0.0005$

6.3 The adaptive integral control

Next the adaptive integral control (4.18) was tested in both the Shepp-Logan and the walnut cases. The figure 6.5 shows the ratio of nonzero coefficients as the iteration progresses for both datasets. The reconstructions can be seen in figures 6.7a and 6.7b. The reconstruction for the Shepp-Logan phantom was achieved in 163 iterations, and the walnut reconstruction in 113 iterations. The relative error of the Shepp-Logan phantom reconstruction was 0.1175.



(a)

(b)

Figure 6.5: The ratio of nonzero wavelet coefficients as the iteration progresses using the adaptive integral control for a) the Shepp-Logan phantom b) the walnut

6.4 The PIDNN control

The PIDNN control was tested for both datasets. For walnut dataset, the learning rate was set as $\eta = 5 \cdot 10^{-12}$. For the Shepp-Logan phantom the learning rate was $\eta = 10^{-7}$. In both cases the weights from input to output weights where set as $w_{1,j} = -1$ and $w_{2,j} = 1$ for $j = P, I, D$. The weights from hidden nodes to the output node was initialized as $w_P = 0.0001$, $w_I = 0.0001$ and $w_D = 0.0001$. The ratio of nonzero coefficients as the iteration progresses can be seen in figure 6.6. The relative error of the Shepp-Logan phantom reconstruction was 0.1258. The table 6.3 shows the comparison of the adaptive integral control and the PIDNN control. The reconstructions can be seen in 6.7c and 6.7d.

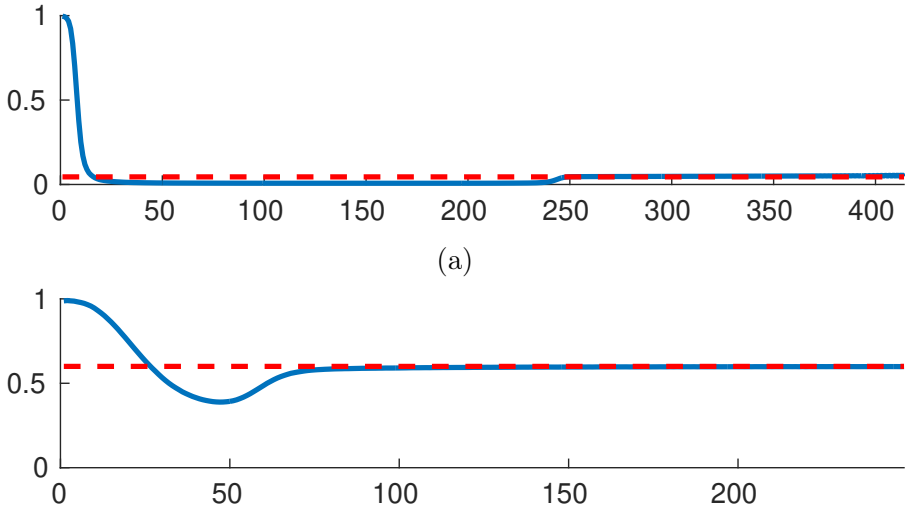
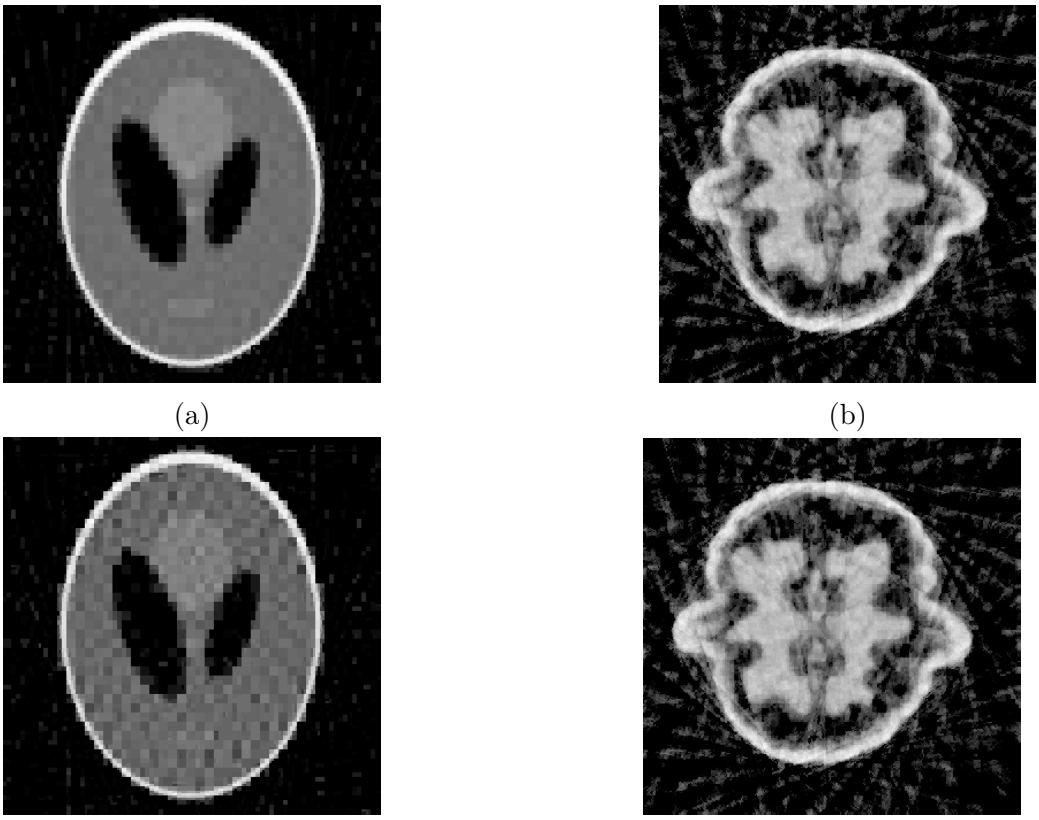


Figure 6.6: The ratio of nonzero wavelet coefficients as the iteration progresses using the PIDNN controller for a) the Shepp-Logan phantom b) the walnut

Table 6.3: Comparison between the adaptive integral control and the PIDNN control

	Absolute error	iter. walnut	iter. S-L	NIAE	PO
Adaptive integral	0.1175	113	163	0.6340	0.3113
PIDNN	0.1258	249	414	6.7858	0.3523



(a) (b)

(c) (d)

Figure 6.7: The reconstructions of the Shepp-Logan phantom and the walnut using (a)(b) the adaptive integral control (c)(d) PIDNN control

6.5 Comparison to other methods

The reconstructions of resolution 164×164 from 120 and 20 projections were computed using the filtered backprojection method, conjugate gradient based Tikhonov regularization method and the integral controlled ISTA. The L-curve was computed using 100 points in the interval $[10^{-16}, 10^{1}]$ and the values of regularization parameters corresponding to the corners were obtained. In the walnut case the value of the Tikhonov regularization parameter for 120 projections was $\alpha_{120} = 0.8774$ and for 20 projections $\alpha_{20} = 1.07 \times 10^{-5}$. For the Shepp-Logan phantom the values were $\alpha_1 20 = 1.2475$ and $\alpha_{20} = 0.0949$. The reconstructions of the walnut using the three methods can be seen in figure 6.8. See the figure 6.9 for reconstructions of the Shepp-Logan phantom from 120 and 20 projection directions using the FBP, Tikhonov regularization and the iterative soft-thresholding algorithm. The relative errors of the reconstructions are presented in the table 6.4.

Table 6.4: The relative errors of the Shepp-Logan phantom reconstructions for the three different reconstruction methods

	120 projections	20 projections
FBP	0.4487	0.7740
Tikhonov	0.2844	0.4906
Adaptive integral	0.3861	0.4493

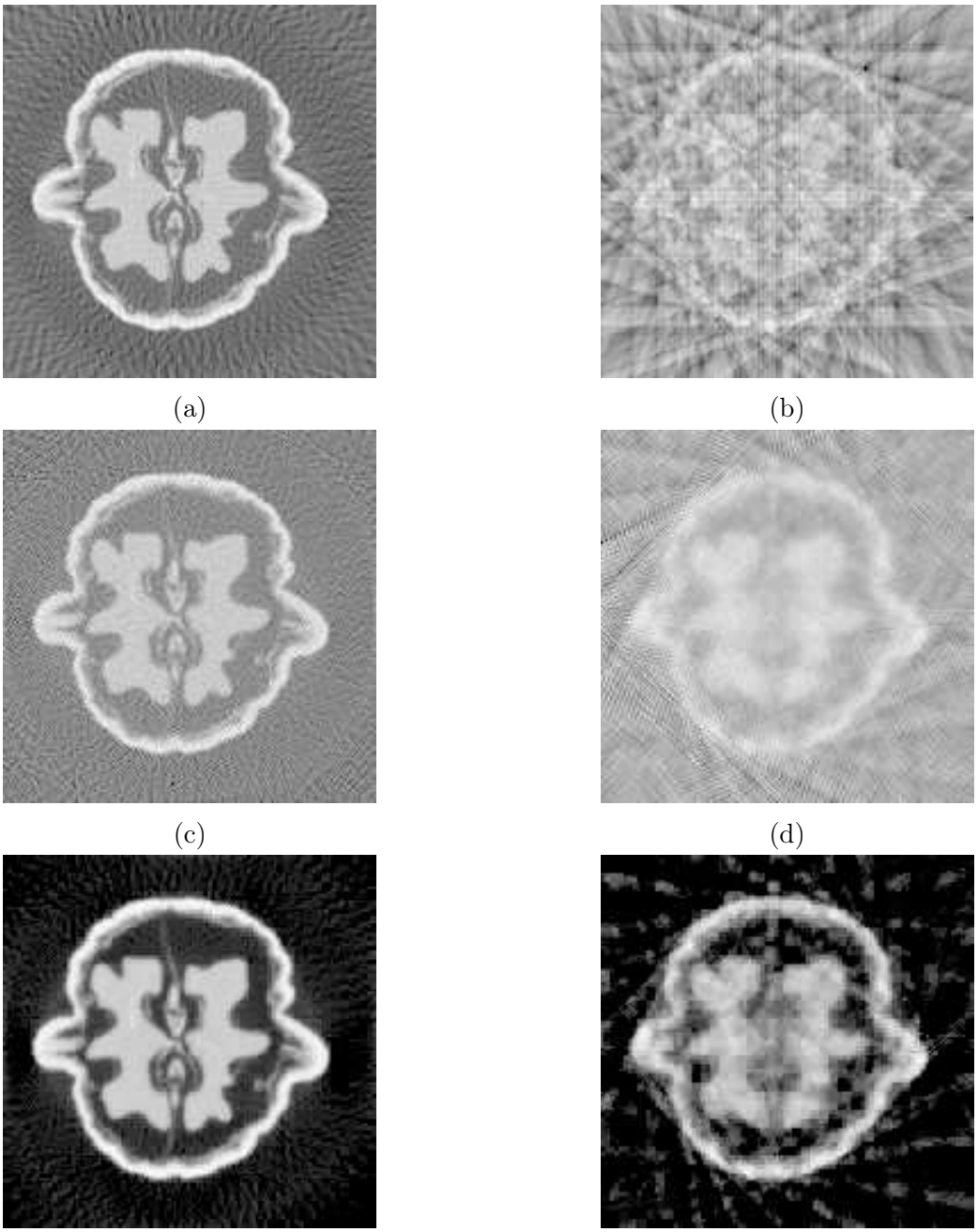
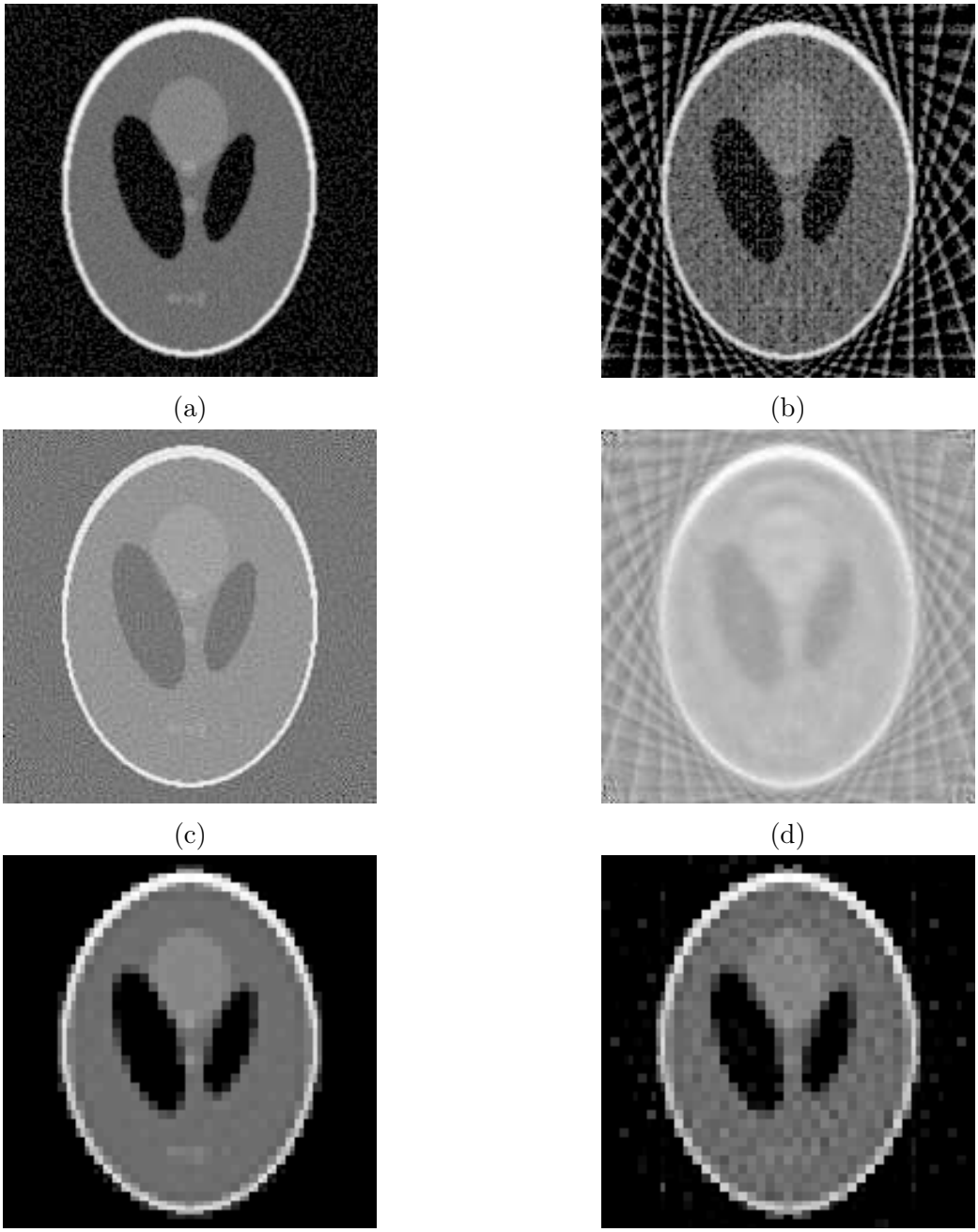


Figure 6.8: The reconstructions of the walnut using three different methods: (a)(b) Filtered backprojection, (c)(d) Tikhonov regularization with the L-curve method (e)(f)adaptive integral controlled iterative soft-thresholding. The reconstructions on the left are from 120 projections, and reconstructions on the right from 20 projections.



(a) (b)

(c) (d)

(e) (f)

Figure 6.9: The reconstructions of the Shepp-Logan fantom with three different methods: (a)(b) the filtered backprojection algorithm (c)(d) Tikhonov regularization with the L-curve method (e)(f) adaptive integral controlled iterative soft-thresholding. On the left are reconstructions from 120 projection directions and on the right from 20 projection directions.

Chapter 7

Discussion

In this thesis we studied sparse data CT-imaging using a sparsity promoting reconstruction method with an automatic regularization parameter selection based on *a priori* known level of sparsity.

Choosing the threshold μ by using PID or related control systems seems to be effective if the controller is properly tuned. In comparison to methods which compute the estimates of the sparsity using multiple values of μ, PID control should be significantly faster. This saves computation time and makes sparsity based parameter selection more applicable to larger scale and in three dimensions. Since PID-controllers are widely used in different industries, many different methods of tuning them exist and thus according to our results could be used in CT-imaging and maybe other inverse problems. Adaptive tuning methods make it possible to create end-user friendly software for sparse CT-imaging.

Pure integral control shows a lot of promise by being very simple to implement and being relatively easily tuned while giving very good results if properly tuned, see figure 6.2. If tuning is too aggressive however, large oscillations appear. On the other hand if K_i is too small the convergence takes a long time. The adaptive tuning method for the integral control works well 6.5. Even though it experiences a large minimum undershoot, it recovers from it quickly. Integral control with the adaptive tuning method seems promising for the actual end user as it requires minimal expertise to use, however, it might not be optimal with other algorithms or data. More study is needed to determine if our choice of p_0 is good enough in other cases.

The results for PI control 6.3 indicate that PI control can work well, and could have some uses when more general type of controller is needed. In the case of wavelet based reconstructions using ISTA, PI control does not perform better than integral control in terms of speed but makes the use of adaptive tuning methods harder. PI-control might be more crucial when controllers are used to control other sparsity promoting algorithms, especially algorithms that are slow to react to changes in the regularization parameter. The velocity control in this case could slow down the control signal reducing ringing. This is shown both by the table 6.1 and the figure

6.3 which show how increasing K_p can be used to eliminate undershoot.

The use of derivative control seems unnecessary in this case and derivative action might do more harm than good, see figure 6.4. As can be seen from the figure 6.4 and the table 6.2, increasing K_d increases the undershoot. The table 6.2 also indicates that using a small K_d could decrease the percentage undershoot slightly in our case the minimum percentage undershoot is achieved using $K_d = 0.00006$, however the negative integral absolute error is still larger than with $K_d = 0$.

The PIDNN controller 4.2 seems to work reasonaby well in the specific examples shown 6.6, however its performance is still far from optimal: the reconstruction of the Shepp-Logan phantom shows significant artefacts inside the phantom even after 400 iterations. In comparison a better reconstruction using the adaptive integral control was obtained in 163 iterations 6.5 and 6.7. Part of the reason is that the network does not employ momentum [22] [23] which could speed up the learning process. The artefacts might also be a sign of the different paths of μ resulting in sligtly different results. Choosing the learning rate is still a problem, however multiple approaches for adaptive learning rates of the backpropagation algorithm have been proposed [24] [25]. Since the adaptive learning rate was not considered here, one should not take the comparison of PIDNN and the adaptive integral control 6.3 too seriously. Artificial neural networks could be useful for other tasks related to the controlling the level of sparsity, such as process model identification to decrease the uncertainty caused by the nonlinearity of the system. Another interesting use for them might be analyzing the reconstructions, giving us additional feedback for the controller. Control methods employing neural networks might be useful when dealing with cases where the system being controlled is highly nonlinear [26] [27].

In terms of the relative error the iterative soft-thresholding algorithm with adaptive integral control performs better than the filtered backprojection method, see the table 6.4 and the reconstructions in the figures 6.9 and 6.8. Our method does not seem to perform significantly better than Tikhonov regularization in terms of relative error, but visual inspection shows that integral controlled ISTA has significantly fewer artefacts when fewer projection angles are used. The small difference between the relative errors of the Tikhonov regularized reconstructions and integral controlled ISTA reconstructions are most probably caused by the very low *a priori* sparsity level, more in depth discussion is included below.

The Shepp-Logan reconstructions in figures 6.7, 6.9e and 6.9f as well as the plots of the sparsity levels of the iterates 6.5a and 6.6a showcase the importance of careful estimation of the *a priori* levels of sparsity. The ratio of nonzero coefficients of the Shepp-Logan phantom were computed using the noise-free target 5.1a which does not have a lot of small details, and therefore very few nonzero coefficients are required to describe it. This becomes a problem as noise and streaky projection artefacts appear, as less nonzero coefficients are available to describe the actual object of interest: the borders of the object become pixelated as seen in 6.7a and especially in the lower resolution reconstructions 6.9e and 6.9f. To avoid the problem one could take the presence of artefacts into account when computing the *a priori* level of sparsity,

therefore increasing both the detail of the object and the amount of artefacts in the resulting reconstruction. This exact problem is likely not as significant in medical imaging or other similar applications as the *a priori* sparsity levels are most probably computed using an atlas of full-angle reconstructions computed from noisy data and therefore sparsity levels obtained are likely higher than in our simulated case. However when fewer projections are used, the problem becomes more significant as the artefacts become more prominent. In such a case the solution could involve using the fact that the pixels outside the object should be zero, therefore leaving a larger portion of the nonzero coefficients to describe the object.

With the adaptive integral control and other resonably good tunings the limiting factor for the speed seems to be the convergence rate of the algorithm being controlled. Therefore to speed up the reconstruction process one would have to use a faster converging algorithm. Similar approaches might work for with other iterative sparsity promoting algorithms such as different variations of the iterative soft thresholding. Methods introduced here should be applicable with other bases and frames besides the Haar wavelet basis. Especially interesting alternative to wavelets are *shearlets* which provide directional awareness. One might also consider such control loops with other than sparsity promoting algorithms, as long as the mapping from the regularization parameter to the measurement being used is sufficiently well behaving.

Even though results here seem to indicate the effectiveness of feedback control systems in CT-imaging, more theoretical and experimental work is needed to find the best control schemes to deal with the difficulties presented by the sparsity promoting algorithms. To proof the convergence of the controlled algorithm, one would have to examine the stability of the closed loop system. Some other variations on the system 4.1 could be considered, such as the more advanced filtering mentioned previously. If the standard PID proves insufficient in some cases one could consider some nonlinear control schemes, such as gain scheduling or neural network based approaches for the task.

Bibliography

[1] F. Natterer, *The Mathematics of Computerized Tomography*. SIAM, 2001.

[2] I. Daubechies, M. Defrise, and C. D. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on pure and applied mathematics*, 2004.

[3] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing, and other inverse problems," *IEEE Journal of Selected Topics in Signal Processing*, 2007.

[4] S. Wright, R. Nowak, and M. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, 2009.

[5] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, 1999.

[6] V. Kolehmainen, M. Lassas, K. Niinimäki, and S. Siltanen, "Sparsity-promoting bayesian inversion," *Inverse Problems*, 2005.

[7] K. Hämäläinen, A. Kallonen, V. Kolehmainen, M. Lassas, K. Niinimäki, and S. Siltanen, "Sparse tomography," *SIAM Journal of Scientific Computing*, 2013.

[8] E. T. Hale, W. Yin, , and Y. Zhang, "Fixed-point continuation for ℓ_1-minimization: Methodology and convergence," *SIAM Journal on Optimization*, 2008.

[9] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design and Tuning*. International Society of Automation, 1995.

[10] J. Siegler and N. Nichols, "Optimum settings for automatic controllers," *Transactions of the ASME*, 1942.

[11] H. Logemann and S. Townley, "Adaptive integral control of time-delay systems," *IEE Proceedings: Control Theory and Applications*, 1997.

[12] Y. P. Huailin Shu, "Pid neural networks for time-delay systems," *Computers and Chemical Engineering*, 2000.

[13] J. Mueller and S. Siltanen, *Linear and Nonlinear Inverse Problems with Practical applications*. SIAM Computational Science & Engineering, 2012.

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Upper Saddle River: Prentice Hall, 2008.

[15] I. Daubechies, *Ten Lectures on Wavelets*. SIAM, 1992.

[16] P. Chen, J. Huang, and X. Zhang, "A primal-dual fixed point algorithm for minimization of the sum of three convex separable functions," *Fixed Point Theory and Applications*, 2016.

[17] Y. Li, K. H. Ang, and G. Chong, "Pid control systems analysis and design - problems, remedies and future directions," *IEEE Control Systems Magazine*, 2006.

[18] M. Garwon, L. Darmadi, F. Urzynicok, G. Barwolff, and R. King, "Adaptive control of separated flows," *European Control Conference (ECC)*, 2003.

[19] K. Hämäläinen, L. Harhanen, A. Kallonen, A. Kujanpää, E. Niemi, and S. Siltanen, "Tomographic x-ray data of a walnut," *arXiv preprint arXiv:1502.04064*, 2015.

[20] C. T. Kelley, *Iterative Methods for Optimization*. SIAM, 1999.

[21] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems*. SIAM, Philadelphia, 1998.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Nature*, 1986.

[23] N. Qian, "On the momentum term in gradient descent learning algorithm," *Neural Networks*, 1999.

[24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, 2011.

[25] V. P. Plagianakos, D. G. Sotiropoulos, and M. N. Vrahatis, "An improved backpropagation method with adaptive learning rate," *Proceedings of the 2nd international conference on circuits, systems and computers*, 1998.

[26] J. Kang, W. Meng, A. Abraham, and H. Liu, "An adaptive pid neural network for complex nonlinear system control," *Neurocomputing*, 2014.

[27] P. Dolezel, "Pid controller design for nonlinear oscillative plants using piecewise linear neural network," *International Conference on Process Control*, 2013.